Magazin de înghețată Pufic

Proiect realizat de Perli Davide Andrea

Contents

[Descriere Model 2](#_Toc188212845)

[Constrângeri 2](#_Toc188212846)

[Descriere Entități 4](#_Toc188212847)

[Descriere relații 5](#_Toc188212848)

[Atribute 6](#_Toc188212849)

[COD SQL : 8](#_Toc188212850)

[Creare secvență 8](#_Toc188212851)

[Creare tabele 8](#_Toc188212852)

[Inserare date în tabele 17](#_Toc188212853)

[Proceduri si functii PLSQL 38](#_Toc188212854)

[Exercițiul 6 (subprogram stocat independent cu 3 tipuri de colectii) 38](#_Toc188212855)

[Exercițiul 7 (subprogram stocat independent 2 tipuri de cursoare, unul fiind parametrizat de celalalt cursor) 41](#_Toc188212856)

[Exercițiul 8 (Functie exceptii no\_data\_found, Too\_Many\_rows, JOIN 3 tabele) 45](#_Toc188212857)

[Exercițiul 9 (PROCEDURA CU 2 PARAMETRI, 2 EXCEPTII PROPRII, JOIN 5 TABELE) 49](#_Toc188212858)

[TRIGGERS 53](#_Toc188212859)

[Exercițiul 10 (LMD la nivel de comanda) 53](#_Toc188212860)

[Exercițiul 11 (LMD la nivel de linie) 60](#_Toc188212861)

[Exercițiul 12 (LDD) 68](#_Toc188212862)

[Pachete 70](#_Toc188212863)

[Exercițiul 13 (2 Tipuri complexe de date, 2 functii, 2 proceduri) - Incomplet 70](#_Toc188212864)

[Diagrama Entitate Relație 76](#_Toc188212865)

[Diagrama Conceptuală 77](#_Toc188212866)

[Scheme relaționale 78](#_Toc188212867)

[Normalizări 78](#_Toc188212868)

# Descriere Model

Un client poate plasa una sau mai multe comenzi într-un magazin și poate să depună mai multe recenzii pentru o comandă (sau niciuna). Un angajat se ocupă de vânzări (relație de tip trei între Angajat, Client și Comenzi), iar fiecare vânzare va avea un timp înregistrat. Utilitatea timpului de vânzare sugerează că o comandă poate fi plasată într-o data diferită de cea a vânzării (spre exemplu: în cazul comenzilor realizate pentru evenimente, ce ar însemna o cantitate mare de înghețată, care poate nu e disponibilă la un singur magazin), de aceea se poate verifica performanța angajaților de a finaliza comenzi într-un timp cât mai scurt de la plasare până la procesare. Comanda poate conține mai multe arome și o cantitate din fiecare, iar fiecare aromă este aprovizionată de un furnizor (pentru o aroma există un singur furnizor, iar un furnizor poate să aprovizioneze mai multe arome). Comanda este pregătită de un angajat care lucrează într-un singur magazin. Toate magazinele au un singur orar, o adresă și cel mult un sponsor. Utilitatea orarului are legătură directă cu un procent de rentabilitate per oră (conceptual numai: reprezintă un procent raportat vânzările per oră ale unui magazin în funcție de un maxim de vânzări per oră) al fiecărui magazin căruia orarul îi este asociat. Un sponsor va stabili alături de firmă o suma de investiție oferită și un număr de evenimente, ce înseamnă obligativitatea firmei de a vinde înghețată la diverse evenimente organizate de sponsor. Sponsorul va avea o adresă. Utilitatea adresei sponsorului este pentru ca firma să poată sugera asocierea magazinelor cu sponsorii în funcție de aria de localizare (dacă sponsorul ar dori promovare locală sau un parteneriat logistic în funcție de zonă).

# Constrângeri

Magazine:

IDMagazin – PRIMARY KEY

NumeMagazin – UNIQUE

RentabilitateOra – o valoare exprimată în procente (între 0.00 și 100.00)

IDAdresa – FOREIGN KEY

IDOrar – FOREIGN KEY

IDSponsor – FOREIGN KEY

Clienti:

IDClient – PRIMARY KEY

Prenume – NOT NULL

Nume – NOT NULL

Comenzi:

IDComanda – PRIMARY KEY

DataComanda – NOT NULL

TipPlata – NOT NULL, o valoare de tip „card” sau „cash”

Sponsor:

IDSponsor – PRIMARY KEY

Nume – NOT NULL

SumaInvestitie – NOT NULL  
IDAdresa – FOREIGN KEY

Angajati:

IDAngajat – PRIMARY KEY

Nume – NOT NULL

Prenume – NOT NULL

DataAngajare – NOT NULL

Salariu – NOT NULL

IDMagazin – FOREIGN KEY

Furnizori:

IDFurnizor – PRIMARY KEY

NumeFurnizor – NOT NULL

Email - NOT NULL

NumarTelefon - NOT NULL

Arome:

IDAroma – PRIMARY KEY

Pret – NOT NULL

NumeAroma – UNIQUE

IDFurnizor – FOREIGN KEY (ON DELETE SET NULL)

Adresa:

IDAdresa – PRIMARY KEY

Vanzari:

(IDClient, IDAngajat, IDComanda) – PRIMARY KEY ale cărei componente sunt FOREIGN KEY

DataVanzare – NOT NULL

OrarMagazine:

IDOrar – PRIMARY KEY

OraDeschidere – NOT NULL

OraInchidere – NOT NULL

Recenzii:

IDRecenzie – PRIMARY KEY

Rating – NOT NULL

IDClient – FOREIGN KEY

AdaugaAroma:

(IDComanda, IDAroma) – PRIMARY KEY ale cărei componente sunt FOREIGN KEY

Cantitate – NOT NULL

# Descriere Entități

1. Magazine

-cheie primară : IDMagazin

-entitate dependentă de Adresa, OrarMagazine, Sponsor

-un magazin are un id, un nume, o locație (adresă), un orar și o rentabilitate per oră (procent vânzări per oră dintr-un maxim înregistrat)

1. Clienti

-cheie primară : IDClient

-entitate independentă

-un client are un id, prenume, nume, email și un număr de telefon

1. Comenzi

-cheie primară : IDComanda

-entitate independentă

-o comandă are un id, o dată și un tip de plată

1. Sponsor

-cheie primară : IDSponsor

-entitate dependentă de Adresa

-un sponsor are un id, o locație (adresă), nume, email, număr de telefon, sumă investiție și număr de evenimente la care participă firma

1. Angajati

-cheie primară : IDAngajat

-entitate dependentă de Magazine

-un angajat are un id, nume, prenume, dată de angajare și salariu

1. Furnizori

-cheie primară : IDFurnizor

-entitate independentă

-un furnizor are un id, nume, email și număr de telefon

1. Arome

-cheie primară : IDAroma

-entitate dependentă de Furnizori

-o aromă are un id, preț și nume

1. Adresa

-cheie primară : IDAdresa

-entitate independentă

-o adresă are un id, oraș și stradă

1. Vanzari

-cheie primară compusă : IDClient, IDComanda, IDMagazin

-entitate dependentă de Clienti, Comenzi și Magazine

-o vanzare are o dată

1. OrarMagazine

-cheie primară : IDOrar

-entitate independentă

-un orar are un id, o oră de deschidere și oră de închidere

1. Recenzii

-cheie primară : IDRecenzie

-entitate dependentă de Clienti

-o recenzie are un id, rating, comentariu și o dată

1. AdaugaAroma

-cheie primară compusă : IDComanda, IDAroma

-entitate dependentă de Comenzi și Arome

-are o cantitate comandată per produs

# Descriere relații

Magazin\_are\_o\_adresa = relație între Magazine și Adresa

Un Magazin poate să aibă o singură Adresa

O Adresa poate să aibă un Magazin sau 0 Magazine

Cardinalitate maximă : 1

Cardinalitate minimă : 1(0)

Magazin\_detine\_un\_orar = relație între Magazine și OrarMagazine

Un Magazin deține un OrarMagazine

Un OrarMagazine poate fi deținut de mai multe Magazine

Cardinalitate maximă : M(1)

Cardinalitate minima : 1

Magazin\_sponsorizat\_de\_sponsor = relație între Magazine și Sponsor

Un Magazin este sponsorizat de un Sponsor sau 0 Sponsori

Un Sponsor poate sponsoriza mai multe Magazine sau un singur Magazin

Cardinalitate maximă : M(1)

Cardinalitate minimă : 1(0)

Sponsor\_are\_o\_adresa = relație între Sponsor și Adresa

Un Sponsor poate să aibă o singură Adresa

O Adresa poate să aibă un Sponsor sau 0 Sponsori

Cardinalitate maximă : 1

Cardinalitate minimă : 1(0)

În\_Magazine\_lucrează\_Angajati = relație între Magazine și Angajati

Într-un Magazin lucrează unul sau mai mulți Angajati

Un Angajat lucrează într-un singur Magazin

Cardinalitate maximă : M(1)

Cardinalitate minimă : 1

Un\_Client\_depune\_Recenzii = relație între Client și Recenzii

Un Client depune mai multe Recenzii sau 0 Recenzii

O Recenzie poate fi depusă de un singur Client

Cardinalitate maximă : 1

Cardinalitate minimă : M(0)

Un\_Furnizor\_aprovizionează\_o\_Aromă = relație între Furnizori și Arome

Un Furnizor aprovizionează o Aroma sau mai multe

O Aroma este furnizată de un singur Furnizor

Cardinalitate maximă : M(1)

Cardinalitate minimă : 1

# Atribute

1. MAGAZINE

IDMagazin # -> cod numeric de tip int

IDAdresa -> cheie străină din tabela Adresa

IDOrar INT -> cheie străină din tabela OrarMagazine

IDSponsor INT -> cheie străină din tabela Sponsor

NumeMagazin -> variabilă de tip varchar2 de lungime 100 de caractere

RentabilitateOra -> variabilă de tip number de lungime 5 cifre, dintre care 2 zecimale

1. CLIENTI

IDClient # -> cod numeric de tip int

Prenume -> variabilă de tip varchar2 de lungime 50 de caractere

Nume -> variabilă de tip varchar2 de lungime 50 de caractere

Email -> variabilă de tip varchar2 de lungime 100 de caractere

NumarTelefon -> variabilă de tip varchar2 de lungime 15 de caractere

1. COMENZI

IDComanda # -> cod numeric de number de 5 cifre

DataComanda -> variabilă de tip timestamp

TipPlata -> variabilă de tip char de lungime de 4 caractere

1. SPONSOR

IDSponsor # -> cod numeric de tip int

IDAdresa -> cheie străină din tabela Adresa

Nume -> variabilă de tip varchar2 de lungime 50 de caractere

Email -> variabilă de tip varchar2 de lungime 100 de caractere

NumarTelefon -> variabilă de tip varchar2 de lungime 15 de caractere

SumaInvestitie -> variabilă de tip number de lungime 6 cifre, dintre care 2 zecimale

NumarEvenimente -> variabilă de tip int

1. ANGAJATI

IDAngajat # -> cod numeric de tip int

IDMagazin -> cheie străină din tabela Magazine

Nume -> variabilă de tip varchar2 de lungime 100 de caractere

Prenume -> variabilă de tip varchar2 de lungime 100 de caractere

DataAngajare -> variabilă de tip date

Salariu DECIMAL(10, 2) -> variabilă de tip decimal

1. FURNIZORI

IDFurnizor # -> cod numeric de tip int

NumeFurnizor -> variabilă de tip varchar2 de lungime 100 de caractere

Email -> variabilă de tip varchar2 de lungime 100 de caractere

NumarTelefon -> variabilă de tip varchar2 de lungime 15 de caractere

1. AROME

IDAroma # -> cod numeric de tip int

IDFurnizor -> cheie străină din tabela Furnizori

Pret -> variabilă de tip decimal

NumeAroma -> variabilă de tip varchar2 de lungime 100 de caractere

1. ADRESA

IDAdresa # -> cod numeric de tip int

Oras -> variabilă de tip varchar2 de lungime 255 de caractere

Strada -> variabilă de tip varchar2 de lungime 255 de caractere

1. VANZARI

IDClient -> cheie străină din tabela Clienti

IDAngajat -> cheie străină din tabela Angajati

IDComanda -> cheie străină din tabela Comenzi

DataVanzare -> variabilă de tip timestamp

1. ORARMAGAZINE

IDOrar # -> cod numeric de tip int

OraDeschidere -> variabilă de tip timestamp

OraInchidere -> variabilă de tip timestamp

1. RECENZII

IDRecenzie # -> cod numeric de tip int

IDClient -> cheie străină din tabela Clienti

Rating -> cod numeric de tip int

Comentariu -> variabilă de tip varchar2 de lungime 255 de caractere

DataRecenzie -> variabilă de tip date

1. ADAUGAAROMA

IDComanda -> cheie străină din tabela Comenzi

IDAroma -> cheie străină din tabela Arome

Cantitate -> variabilă de tip int

# COD SQL :

## Creare secvență

CREATE SEQUENCE SecventaComanda

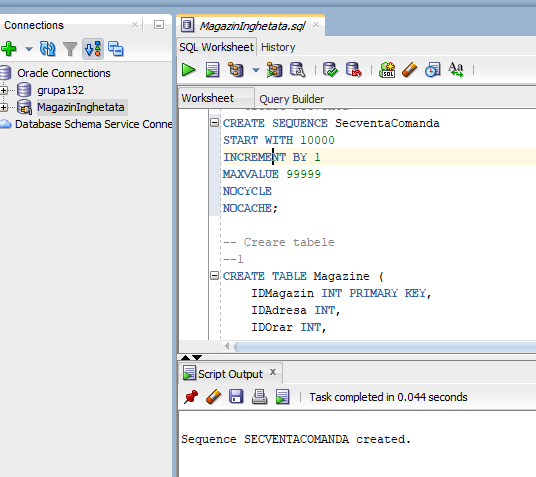
START WITH 10000

INCREMENT BY 1

MAXVALUE 99999

NOCYCLE

NOCACHE;



## Creare tabele

/\*DELETE FROM AdaugaAroma;

DELETE FROM Recenzii;

DELETE FROM Vanzari;

DELETE FROM Angajati;

DELETE FROM Comenzi;

DELETE FROM Arome;

DELETE FROM Furnizori;

DELETE FROM Clienti;

DELETE FROM Magazine;

DELETE FROM Sponsor;

DELETE FROM OrarMagazine;

DELETE FROM Adresa;

DROP SEQUENCE SecventaComanda;

DROP TABLE Vanzari CASCADE CONSTRAINTS;

DROP TABLE Comenzi CASCADE CONSTRAINTS;

DROP TABLE Clienti CASCADE CONSTRAINTS;

DROP TABLE Sponsor CASCADE CONSTRAINTS;

DROP TABLE Magazine CASCADE CONSTRAINTS;

DROP TABLE Furnizori CASCADE CONSTRAINTS;

DROP TABLE Arome CASCADE CONSTRAINTS;

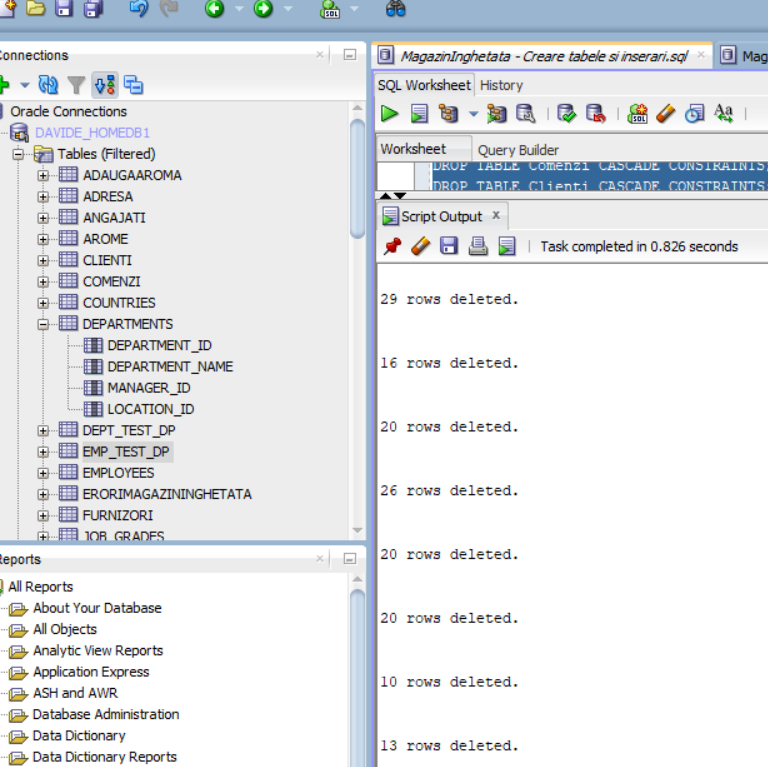
DROP TABLE Angajati CASCADE CONSTRAINTS;

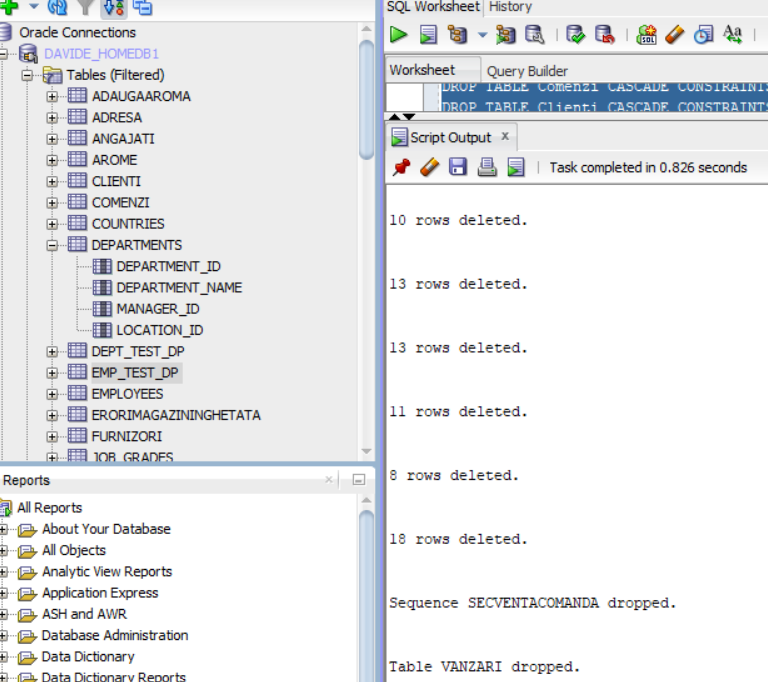
DROP TABLE OrarMagazine CASCADE CONSTRAINTS;

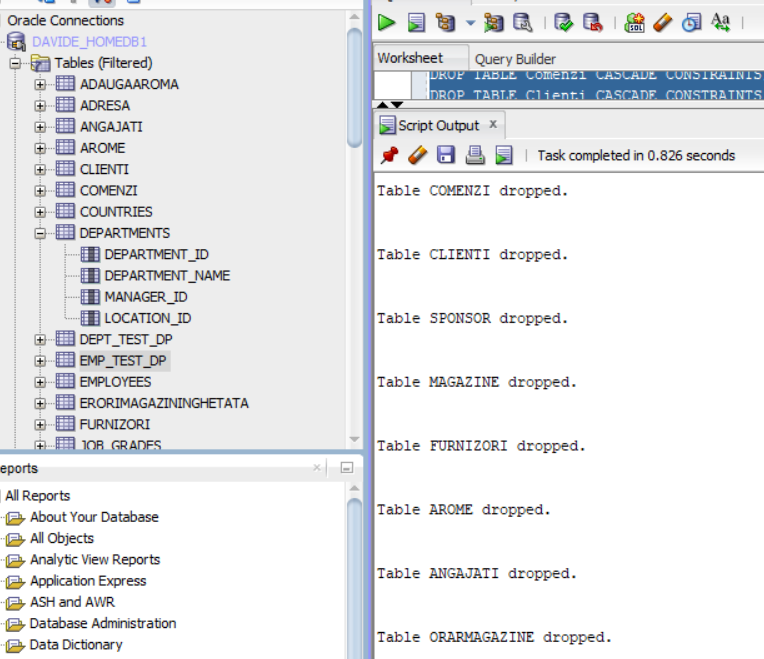
DROP TABLE Recenzii CASCADE CONSTRAINTS;

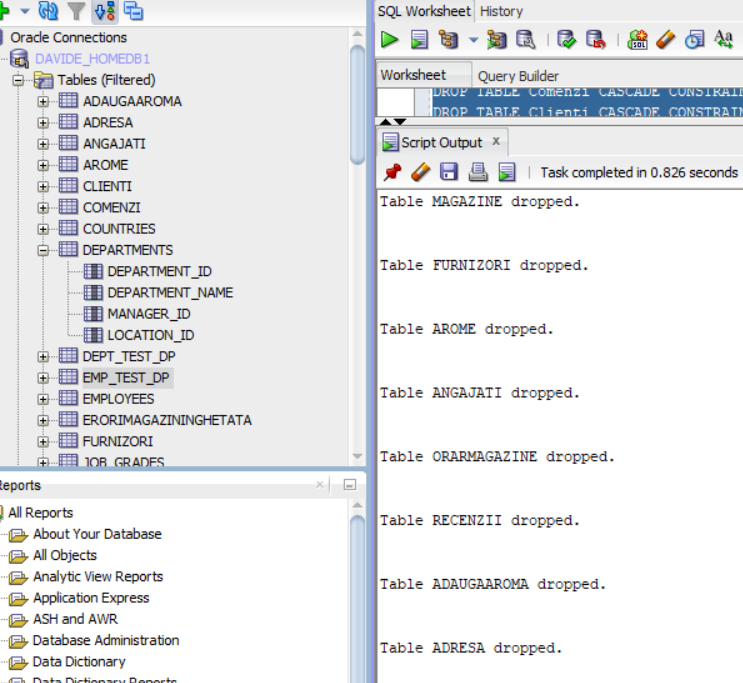
DROP TABLE AdaugaAroma CASCADE CONSTRAINTS;

DROP TABLE Adresa CASCADE CONSTRAINTS;\*/









-- Creare secventa

CREATE SEQUENCE SecventaComanda

START WITH 10000

INCREMENT BY 1

MAXVALUE 99999

NOCYCLE

NOCACHE;

-- Creare tabele

--1

CREATE TABLE Clienti (

    IDClient INT PRIMARY KEY,

    Prenume VARCHAR2(50) NOT NULL,

    Nume VARCHAR2(50) NOT NULL,

    Email VARCHAR2(100),

    NumarTelefon VARCHAR2(15)

);

--2

CREATE TABLE Comenzi (

    IDComanda NUMBER(5) PRIMARY KEY,

    DataComanda TIMESTAMP NOT NULL,

    TipPlata CHAR(4) NOT NULL,

    CONSTRAINT Plata CHECK(TipPlata = 'card' OR TipPlata = 'cash')

);

--3

CREATE TABLE Adresa (

    IDAdresa INT PRIMARY KEY,

    Oras VARCHAR(255),

    Strada VARCHAR(255)

);

--4

CREATE TABLE Sponsor (

    IDSponsor INT PRIMARY KEY,

    IDAdresa INT,

    Nume VARCHAR2(50) NOT NULL,

    Email VARCHAR2(100),

    NumarTelefon VARCHAR2(15),

    SumaInvestitie NUMBER(6, 2) NOT NULL,

    NumarEvenimente INT,

    FOREIGN KEY (IDAdresa) REFERENCES Adresa (IDAdresa)

);

--5

CREATE TABLE OrarMagazine (

    IDOrar INT PRIMARY KEY,

    OraDeschidere TIMESTAMP NOT NULL,

    OraInchidere TIMESTAMP NOT NULL

);

--6

CREATE TABLE Magazine (

    IDMagazin INT PRIMARY KEY,

    IDAdresa INT,

    IDOrar INT,

    IDSponsor INT,

    NumeMagazin VARCHAR2(100) UNIQUE,

    RentabilitateOra NUMBER(5, 2),

    CONSTRAINT ConstraintProcent CHECK(RentabilitateOra > 0 AND RentabilitateOra < 100),

    FOREIGN KEY (IDAdresa) REFERENCES Adresa (IDAdresa),

    FOREIGN KEY (IDOrar) REFERENCES OrarMagazine(IDOrar),

    FOREIGN KEY (IDSponsor) REFERENCES  Sponsor(IDSponsor)

);

--7

CREATE TABLE Angajati (

    IDAngajat INT PRIMARY KEY,

    IDMagazin INT,

    Nume VARCHAR2(100) NOT NULL,

    Prenume VARCHAR2(100) NOT NULL,

    DataAngajare DATE NOT NULL,

    Salariu DECIMAL(10, 2) NOT NULL,

    FOREIGN KEY (IDMagazin) REFERENCES Magazine(IDMagazin)

);

--8

CREATE TABLE Furnizori (

    IDFurnizor INT PRIMARY KEY,

    NumeFurnizor VARCHAR2(100) NOT NULL,

    Email VARCHAR2(100),

    NumarTelefon VARCHAR2(15)

);

--9

CREATE TABLE Arome (

    IDAroma INT PRIMARY KEY,

    IDFurnizor INT,

    Pret DECIMAL(10, 2) NOT NULL,

    NumeAroma VARCHAR2(100) UNIQUE,

    FOREIGN KEY (IDFurnizor) REFERENCES Furnizori (IDFurnizor) ON DELETE SET NULL

);

--10

CREATE TABLE Vanzari (

    IDClient INT,

    IDAngajat INT,

    IDComanda INT,

    DataVanzare TIMESTAMP NOT NULL,

    PRIMARY KEY (IDClient, IDAngajat, IDComanda),

    FOREIGN KEY (IDClient) REFERENCES Clienti(IDClient),

    FOREIGN KEY (IDAngajat) REFERENCES Angajati(IDAngajat),

    FOREIGN KEY (IDComanda) REFERENCES Comenzi(IDComanda) ON DELETE CASCADE

);

--11

CREATE TABLE Recenzii (

    IDRecenzie INT PRIMARY KEY,

    IDClient INT,

    Rating INT NOT NULL,

    Comentariu VARCHAR2(255),

    DataRecenzie DATE,

    FOREIGN KEY (IDClient) REFERENCES Clienti (IDClient)

);

--12

CREATE TABLE AdaugaAroma (

    IDComanda INT,

    IDAroma INT,

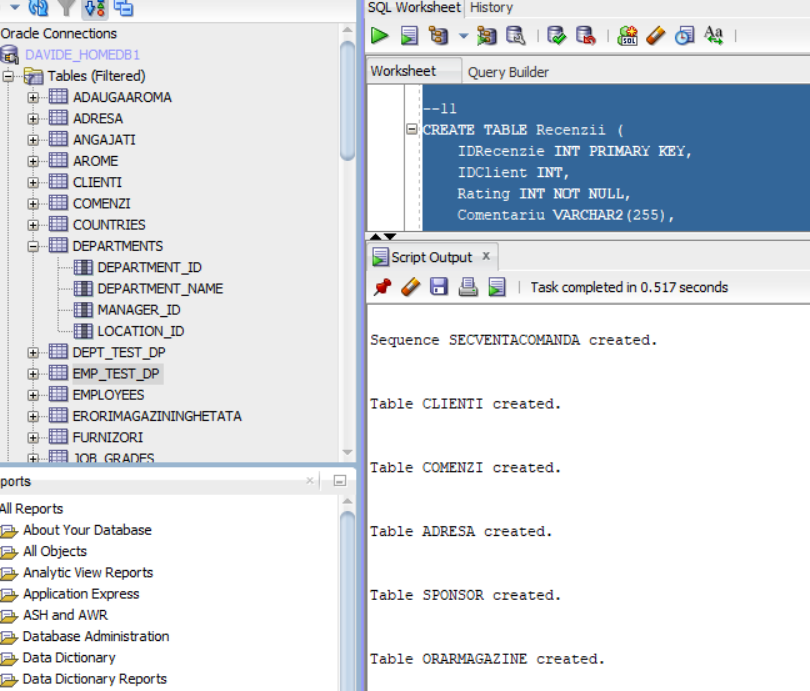
    Cantitate INT NOT NULL,

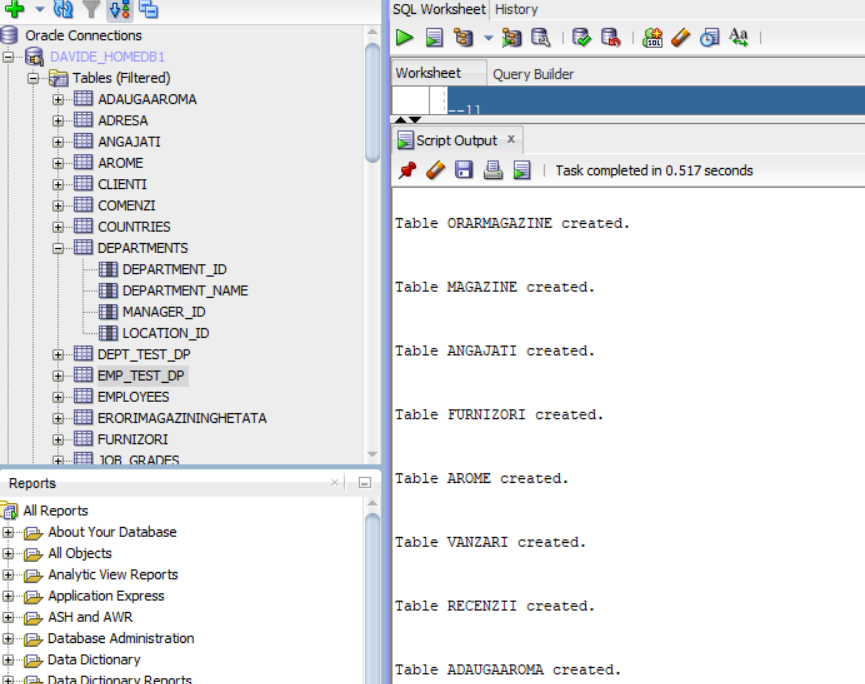
    PRIMARY KEY (IDComanda, IDAroma),

    FOREIGN KEY (IDComanda) REFERENCES Comenzi (IDComanda) ON DELETE CASCADE,

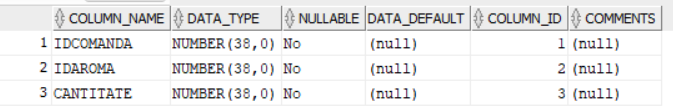
    FOREIGN KEY (IDAroma) REFERENCES Arome (IDAroma)

);

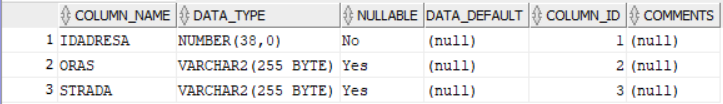




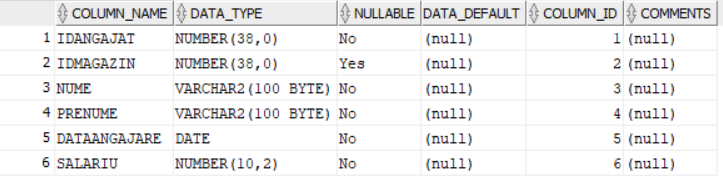
ADAUGAAROMA



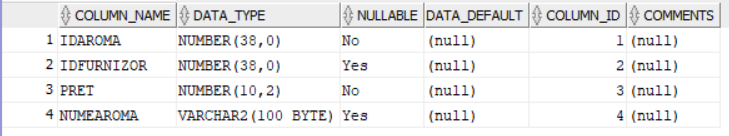
ADRESA



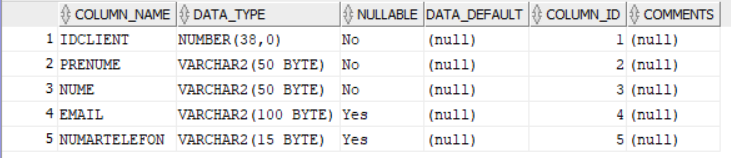
ANGAJATI



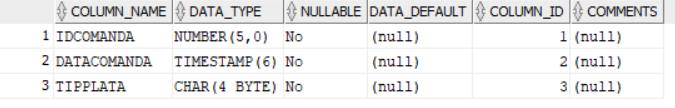
AROME



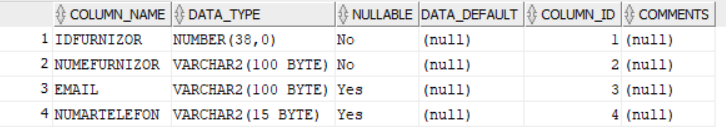
CLIENTI



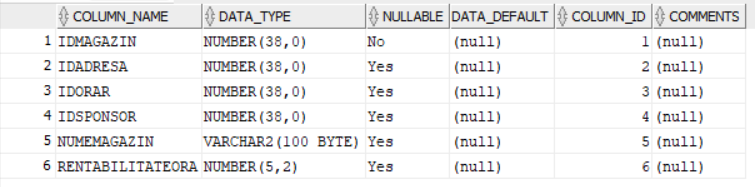
COMENZI



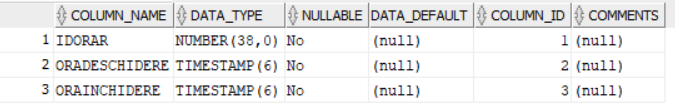
FURNIZORI



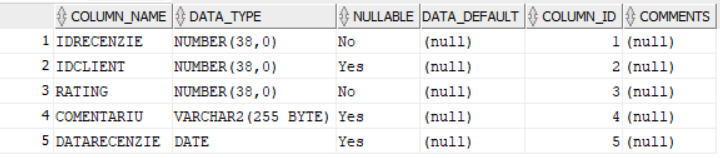
MAGAZINE



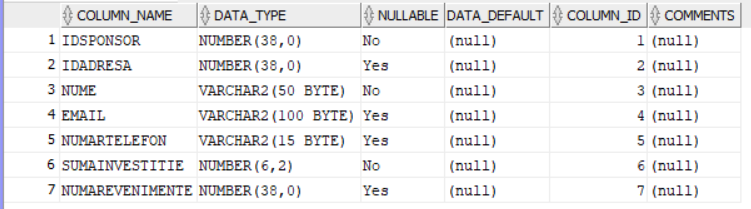
ORARMAGAZINE



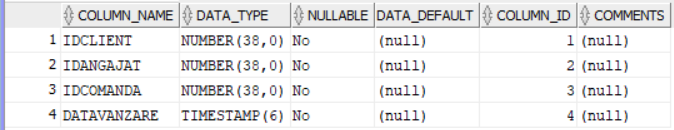
RECENZII



SPONSOR



VANZARI



## Inserare date în tabele

-- Inserare date in tabele

--1 Inserare date in Clienti

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (1, 'Alin', 'George', 'alin.george@gmail.com', '1234567890');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (2, 'Teo', 'Vas', 'teo.vasile@gmail.com', '0987654321');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (3, 'Maria', 'Beni', 'maria.beni@gmail.com', '0687645321');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (4, 'Ana', 'Vasilescu', 'ana.vasilescu@gmail.com', '0775101171');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (5, 'Robi', 'Nico', 'robi.nico@gmail.com', '0216674478');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (6, 'Andrei', 'Paun', 'andrei.paun@gmail.com', '0217654321');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (7, 'George', 'Bob', 'george.bob@gmail.com', '0887654321');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (8, 'Gabi', 'Paki', 'gabi.paki@gmail.com', '087240521');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (9, 'Gabriel', 'Parintele', 'gabriel.parintele@gmail.com', '0314375461');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (10, 'Ionut', 'Pacate', 'ionut.pacate@gmail.com', '0354654321');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (11, 'Teodor', 'Ene', 'teodor.ene@gmail.com', '0259954624');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (12, 'Andra', 'Andreescu', 'andra.andreescu@gmail.com', '078101178');

INSERT INTO Clienti (IDClient, Prenume, Nume, Email, NumarTelefon)

VALUES (13, 'Melisa', 'Marcel', 'melisa.marcel@gmail.com', '0554121363');

--2 Inserare date in Comenzi

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-05-01 14:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-05-02 15:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-06-02 13:42:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2022-08-24 15:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-07-30 11:35:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2022-01-02 17:41:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-08-21 14:48:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-03-09 11:39:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-12-02 10:28:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-11-29 19:25:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-09-25 18:18:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-01-12 20:30:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-06-22 20:55:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-05-19 09:55:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-02-09 11:23:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-04-14 12:35:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-03-11 17:21:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2023-11-17 20:50:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-05-13 14:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'card');

INSERT INTO Comenzi (IDComanda, DataComanda, TipPlata)

VALUES (SecventaComanda.nextval, TO\_DATE('2024-09-09 12:45:00', 'YYYY-MM-DD HH24:MI:SS'), 'cash');

--3 Inserare date in Adresa

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (1, 'Bulevardul Grigorescu 3', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (2, 'Bulevardul Decebal 2', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (3, 'Strada Lotrioarei 4', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (4, 'Intrarea Odobesti 8', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (5, 'Bulevardul Traian 9', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (6, 'Strada Bordesti 4', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (7, 'Bulevardul Victoriei 23', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (8, 'Bulevardul Kiselef 17', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (9, 'Bulevardul Roseti 7', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (10, 'Bulevardul Unirii 1', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (11, 'Strada Livezilor 18', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (12, 'Soseaua Oltenitei 6', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (13, 'Soseaua Pajura 34', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (14, 'Bulevardul Roseti 34', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (15, 'Soseaua Pajura 5', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (16, 'Bulevardul Grigorescu 34', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (17, 'Bulevardul Decebal 17', 'Bucuresti');

INSERT INTO Adresa (IDAdresa, Strada, Oras)

VALUES (18, 'Strada Lunca Bradului 3', 'Bucuresti');

--4 Inserare date in Sponsor

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (1, 5, 'DairyQueen', 'dairy.queen@gmail.com', '0880101170', 3455.99, 3);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (2, 3, 'Alpro', 'alpro@gmail.com', '0770101170', 5670.20, 4);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (3, 14, 'Olympus', 'olympus@gmail.com', '021101170', 4320.50, 6);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (4, 17, 'Zuzu', 'zuzu@gmail.com', '0880187170', 6779.99, 3);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (5, 6, 'Heidi', 'heidi@gmail.com', '0310107870', 7000.50, 6);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (6, 12, 'Milka', 'milka@gmail.com', '0550101170', 9100.30, 6);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (7, 2, 'Michelin', 'michelin@gmail.com', '0450101170', 5000.60, 3);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (8, 15, 'PlusINC', 'plus@gmail.com', '0225434587', 3029.99, 2);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (9, 16, 'Ikea', 'ikea@gmail.com', '0334101170', 7499.99, 6);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (10, 18, 'XXLUX', 'xxlux@gmail.com', '0775801170', 9350.30, 3);

INSERT INTO Sponsor (IDSponsor, IDAdresa, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente)

VALUES (11, 16, 'Decathlon', 'decathlon@gmail.com', '0780254170', 2345.99, 2);

--5 Inserare date in tabela OrarMagazine

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (1, TO\_DATE('08:30:00', 'HH24:MI:SS'), TO\_DATE('17:25:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (2, TO\_DATE('09:00:00', 'HH24:MI:SS'), TO\_DATE('23:00:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (3, TO\_DATE('10:00:00', 'HH24:MI:SS'), TO\_DATE('14:00:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (4, TO\_DATE('07:30:00', 'HH24:MI:SS'), TO\_DATE('19:30:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (5, TO\_DATE('09:30:00', 'HH24:MI:SS'), TO\_DATE('18:30:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (6, TO\_DATE('07:00:00', 'HH24:MI:SS'), TO\_DATE('21:30:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (7, TO\_DATE('08:30:00', 'HH24:MI:SS'), TO\_DATE('21:30:00', 'HH24:MI:SS'));

INSERT INTO OrarMagazine (IDOrar, OraDeschidere, OraInchidere)

VALUES (8, TO\_DATE('10:30:00', 'HH24:MI:SS'), TO\_DATE('22:30:00', 'HH24:MI:SS'));

--6 Inserare date in Magazine

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (1, 1, 7, 11, 'Pufic 1', 70.90);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (2, 2, 5, 11, 'Pufic 2', 57.50);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (3, 3, 8, 3, 'Pufic 3', 49.99);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (4, 4, 8, 1, 'Pufic 4', 23.79);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (5, 5, 8, 5, 'Pufic 5', 78.80);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (6, 6, 1, 6, 'Pufic 6', 99.99); -- conceptual trebuie sa existe un maxim

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (7, 7, 3, 8, 'Pufic 7', 75.81);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (8, 8, 2, 2, 'Pufic 8', 68.99);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (9, 9, 3, 7, 'Pufic 9', 95.55);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (10, 10, 4, 4, 'Pufic 10', 57.89);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (11, 11, 7, 2, 'Pufic 11', 81.23);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (12, 12, 6, 10, 'Pufic 12', 87.47);

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (13, 13, 5, 9, 'Pufic 13', 66.11);

--7 Inserare date pentru Angajati

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (1, 1, 'Popescu', 'Ion', TO\_DATE('2020-01-15', 'YYYY-MM-DD'), 3500.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (2, 1, 'Popa', 'Iooan', TO\_DATE('2020-03-18', 'YYYY-MM-DD'), 3300.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (3, 1, 'Ionescu', 'Maria', TO\_DATE('2019-05-20', 'YYYY-MM-DD'), 4000.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (4, 2, 'Ioan', 'Antonio', TO\_DATE('2018-12-20', 'YYYY-MM-DD'), 4000.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (5, 3, 'Constantinescu', 'Ana', TO\_DATE('2021-03-10', 'YYYY-MM-DD'), 3200.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (6, 10, 'Constantin', 'Teo', TO\_DATE('2021-02-10', 'YYYY-MM-DD'), 3210.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (7, 3, 'Dumitrescu', 'Mihai', TO\_DATE('2018-11-05', 'YYYY-MM-DD'), 4200.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (8, 4, 'Dumitru', 'Matei', TO\_DATE('2018-09-04', 'YYYY-MM-DD'), 4100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (9, 2, 'Armin', 'Matei', TO\_DATE('2024-05-14', 'YYYY-MM-DD'), 2200.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (10, 1, 'Anastasia', 'Maria', TO\_DATE('2024-01-10', 'YYYY-MM-DD'), 2300.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (11, 6, 'Giunca', 'Andrei', TO\_DATE('2024-05-22', 'YYYY-MM-DD'), 2000.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (12, 6, 'Gheorge', 'Roberto', TO\_DATE('2022-07-25', 'YYYY-MM-DD'), 2800.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (13, 9, 'Barca', 'Stefan', TO\_DATE('2023-10-03', 'YYYY-MM-DD'), 2700.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (14, 7, 'Bankai', 'Mihai', TO\_DATE('2023-07-30', 'YYYY-MM-DD'), 2750.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (15, 8, 'Stefanescu', 'Andi', TO\_DATE('2022-09-14', 'YYYY-MM-DD'), 3000.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (16, 5, 'Stamate', 'Andrei', TO\_DATE('2022-02-21', 'YYYY-MM-DD'), 3100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (17, 6, 'Scarlat', 'Ana', TO\_DATE('2021-07-23', 'YYYY-MM-DD'), 4000.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (18, 9, 'Voinescu', 'Anastasia', TO\_DATE('2020-08-27', 'YYYY-MM-DD'), 4300.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (19, 11, 'Enache', 'Maria', TO\_DATE('2020-11-17', 'YYYY-MM-DD'), 4200.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (20, 10, 'Enitei', 'Andreea', TO\_DATE('2020-10-19', 'YYYY-MM-DD'), 4100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (21, 11, 'Ridu', 'Monica', TO\_DATE('2019-12-08', 'YYYY-MM-DD'), 4300.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (22, 11, 'Tunaru', 'Monica', TO\_DATE('2019-08-05', 'YYYY-MM-DD'), 4100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (23, 9, 'Enica', 'Raluca', TO\_DATE('2023-08-25', 'YYYY-MM-DD'), 3100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (24, 12, 'Afina', 'Ramona', TO\_DATE('2023-08-20', 'YYYY-MM-DD'), 3100.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (25, 3, 'Anapol', 'Denisa', TO\_DATE('2022-09-30', 'YYYY-MM-DD'), 3700.00);

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (26, 13, 'Ceausu', 'Dakia', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);

--8 Inserare date in Furnizori

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (1, 'Chocolate Factory', 'chocolate.factory@gmail.com', '1234567890');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (2, 'Carrefour', 'carrefour@gmail.com', '9876543210');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (3, 'Lidl', 'lidl@gmail.com', '5558889999');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (4, 'MegaImage', 'megaimage@gmail.com', '1548889939');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (5, 'Oreo', 'oreo@gmail.com', '07558889987');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (6, 'GusturiRomanesti', 'gusturiromanesti@gmail.com', '03225689999');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (7, 'Mizo', 'mizo@gmail.com', '355779999');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (8, 'ChocoMix', 'chocomix@gmail.com', '0658389979');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (9, 'CocaCola', 'cocacola@gmail.com', '2158866499');

INSERT INTO Furnizori (IDFurnizor, NumeFurnizor, Email, NumarTelefon)

VALUES (10, 'Fruitty', 'fruitty@gmail.com', '041548523');

--9 Inserare date in Arome

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (1, 1, 5.50, 'Fistic');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (2, 1, 2.50, 'Ciocolata');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (3, 1, 3.45, 'Cookies and Cream');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (4, 2, 1.45, 'Bubblegum');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (5, 4, 5.35, 'Cirese');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (6, 1, 6.75, 'Kinder');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (7, 7, 3.00, 'Banane');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (8, 5, 5.65, 'Oreo');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (9, 10, 2.75, 'Caise');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (10, 10, 3.25, 'Menta');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (11, 2, 3.00, 'Alune');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (12, 6, 1.65, 'Portocale');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (13, 8, 4.15, 'Caramel');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (14, 10, 3.00, 'Lamaie');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (15, 10, 3.25, 'Fructe de padure');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (16, 7, 2.00, 'Vanilie');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (17, 3, 3.00, 'Zmeura');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (18, 4, 6.55, 'Nuci si smochine');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (19, 9, 4.55, 'Coca Cola');

INSERT INTO Arome (IDAroma, IDFurnizor, Pret, NumeAroma)

VALUES (20, 8, 4.40, 'Triple Choco');

--10 Inserare date in Vanzari

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (1, 21, 10000, TO\_DATE('2023-05-01 15:00:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 30 min

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (1, 1, 10001, TO\_DATE('2023-05-04 15:45:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 2 zile

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (1, 3, 10002, TO\_DATE('2023-06-02 13:42:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (2, 12, 10003, TO\_DATE('2022-08-27 17:00:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 3 zile

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (2, 12, 10004, TO\_DATE('2024-07-31 12:35:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 1 zi

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (3, 11, 10005, TO\_DATE('2022-01-02 18:41:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 1 ora

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (3, 14, 10006, TO\_DATE('2024-08-23 14:48:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 2 zile

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (4, 12, 10007, TO\_DATE('2024-03-09 11:39:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (5, 13, 10008, TO\_DATE('2023-12-03 10:28:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 1 zi

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (6, 3, 10009, TO\_DATE('2024-11-29 20:25:00', 'YYYY-MM-DD HH24:MI:SS'));-- dupa 1 ora

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (7, 6, 10010, TO\_DATE('2024-09-25 18:18:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (7, 7, 10011, TO\_DATE('2024-01-12 20:30:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (8, 21, 10012, TO\_DATE('2024-06-22 20:55:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (9, 20, 10013, TO\_DATE('2024-05-19 09:55:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (10, 9, 10014, TO\_DATE('2023-02-09 11:23:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (11, 25, 10015, TO\_DATE('2023-04-14 12:35:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (12, 1, 10016, TO\_DATE('2023-03-11 17:21:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (12, 2, 10017, TO\_DATE('2023-11-17 20:50:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (13, 25, 10018, TO\_DATE('2024-05-13 14:45:00', 'YYYY-MM-DD HH24:MI:SS'));

INSERT INTO Vanzari (IDClient, IDAngajat, IDComanda, DataVanzare)

VALUES (13, 7, 10019, TO\_DATE('2024-09-09 12:45:00', 'YYYY-MM-DD HH24:MI:SS'));

--11 Inserare date in tabela Recenzii

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (1, 6, 4, 'Foarte multumit de calitatea produsului!', TO\_DATE('2023-06-01', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (2, 3, 2, 'Produsul a fost OK, dar livrarea a  nt rziat', TO\_DATE('2023-06-02', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (3, 7, 3, NULL, TO\_DATE('2023-08-02', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (4, 2, 5, 'Produsul a fost bun, dar cam scump', TO\_DATE('2023-09-24', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (5, 5, 3, NULL, TO\_DATE('2021-09-30', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (6, 4, 4, 'Foarte buna inghetata', TO\_DATE('2021-03-02', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (7, 8, 5, NULL, TO\_DATE('2022-10-21', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (8, 4, 5, 'O inghetata delicioasa', TO\_DATE('2024-04-09', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (9, 5, 5, 'Una dintre cele mai bune inghetate', TO\_DATE('2024-01-02', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (10, 5, 4, 'O sa ma intorc cu siguranta', TO\_DATE('2023-12-29', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (11, 2, 2, 'Prea scump', TO\_DATE('2023-10-25', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (12, 12, 5, 'Un gust unic si autentic', TO\_DATE('2024-02-12', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (13, 5, 4, 'Extraordinar', TO\_DATE('2023-07-22', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (14, 12, 4, NULL, TO\_DATE('2023-07-22', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (15, 3, 3, NULL, TO\_DATE('2023-07-22', 'YYYY-MM-DD'));

INSERT INTO Recenzii (IDRecenzie, IDClient, Rating, Comentariu, DataRecenzie)

VALUES (16, 13, 5, 'Recomand inghetata de fistic, e cea mai buna', TO\_DATE('2023-07-22', 'YYYY-MM-DD'));

--12 Inserare date in tabela AdaugaAroma

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10000, 1, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10000, 11, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10001, 2, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10002, 5, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10002, 17, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10003, 3, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10004, 6, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10005, 9, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10005, 7, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10005, 2, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10006, 7, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10007, 8, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10008, 4, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10009, 18, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10010, 20, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10010, 19, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10010, 16, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10011, 16, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10012, 6, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10013, 1, 3);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10014, 13, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10015, 11, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10015, 1, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10016, 16, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10016, 20, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10016, 6, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10017, 13, 1);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10018, 17, 2);

INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

VALUES (10018, 10, 1);

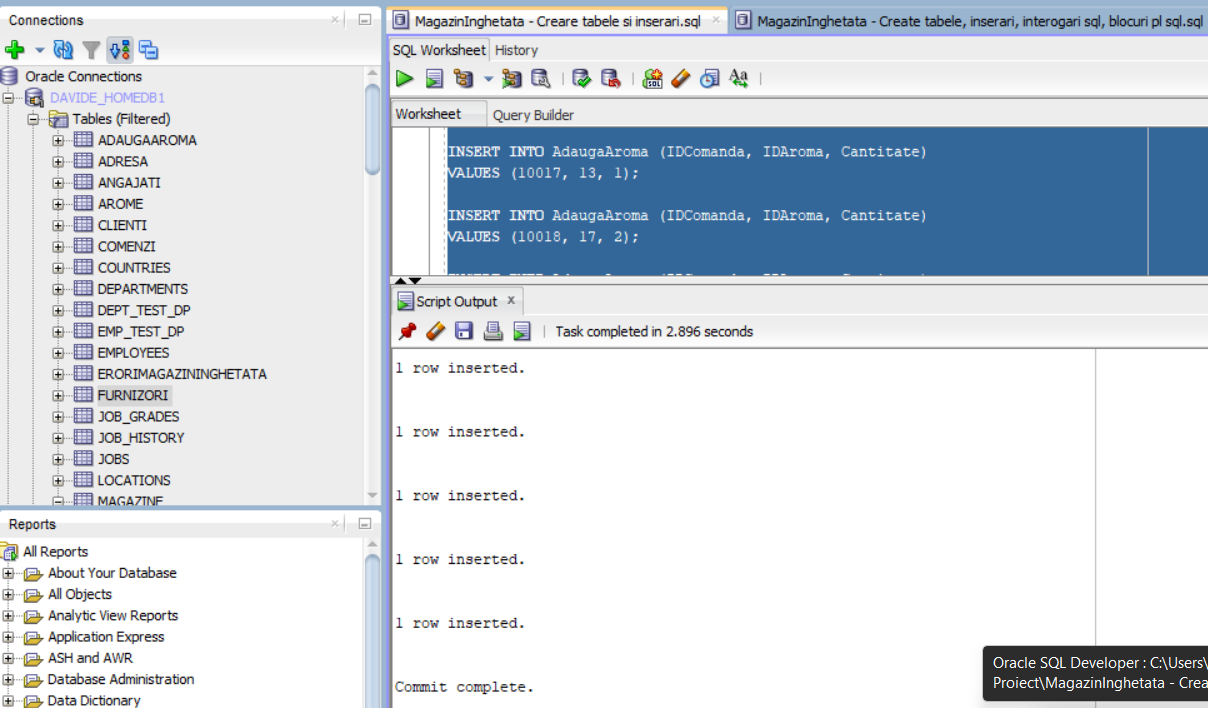
INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

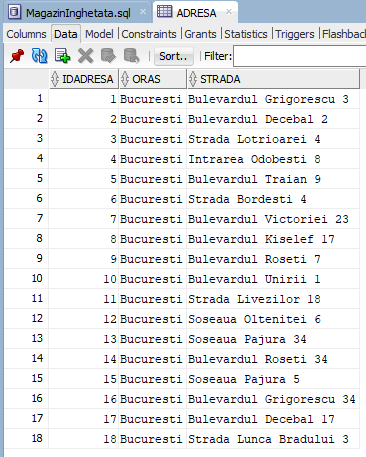
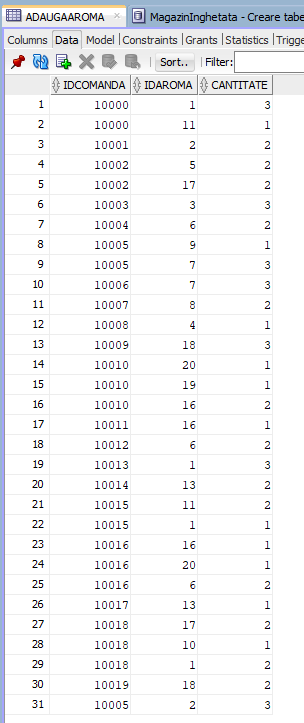
VALUES (10018, 1, 2);

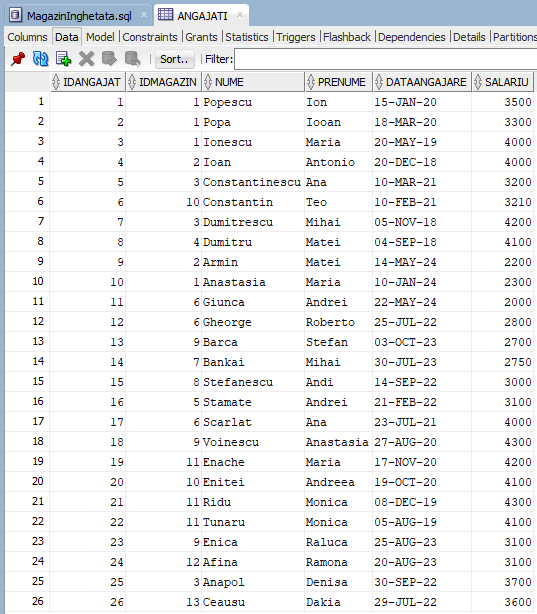
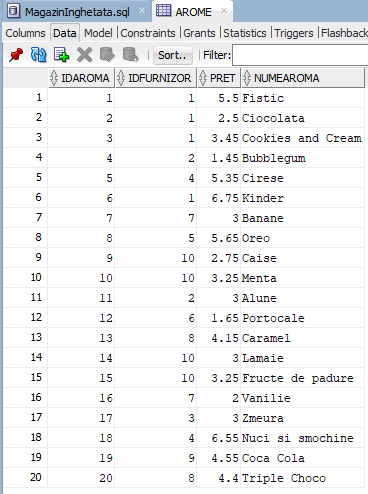
INSERT INTO AdaugaAroma (IDComanda, IDAroma, Cantitate)

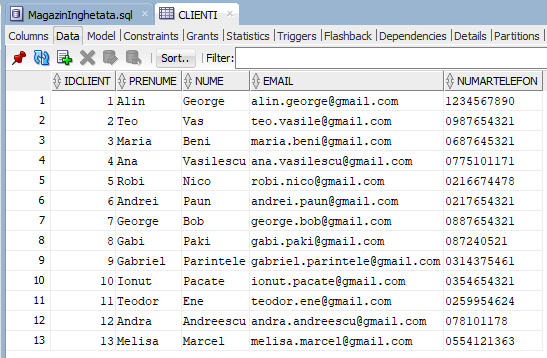
VALUES (10019, 18, 2);

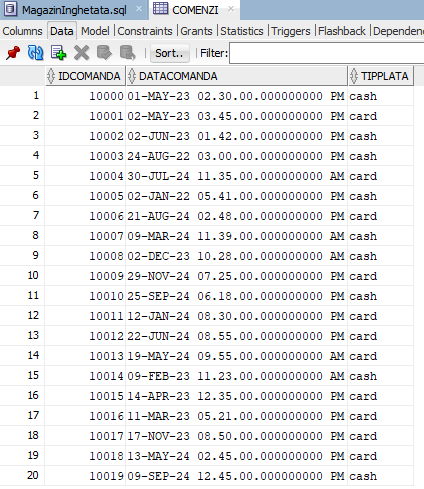
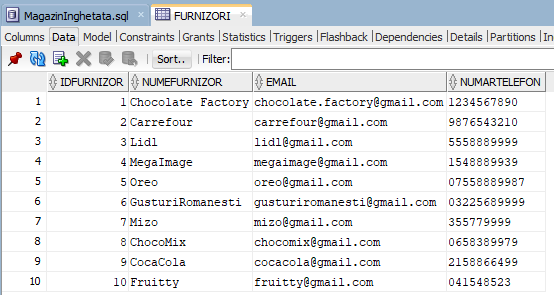
COMMIT;

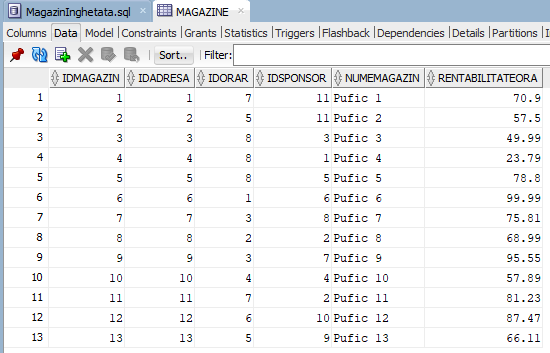


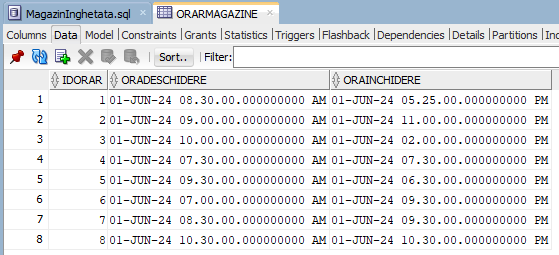


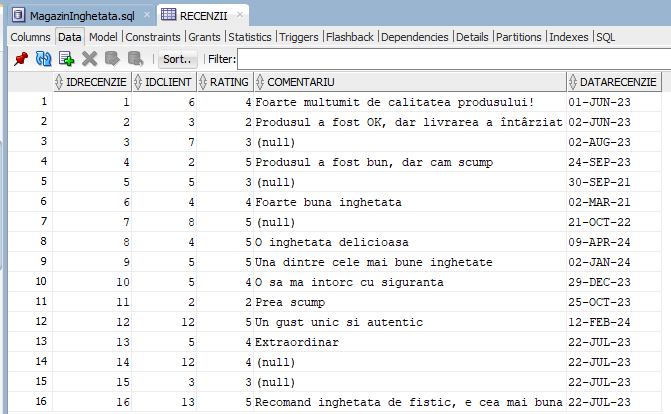
 

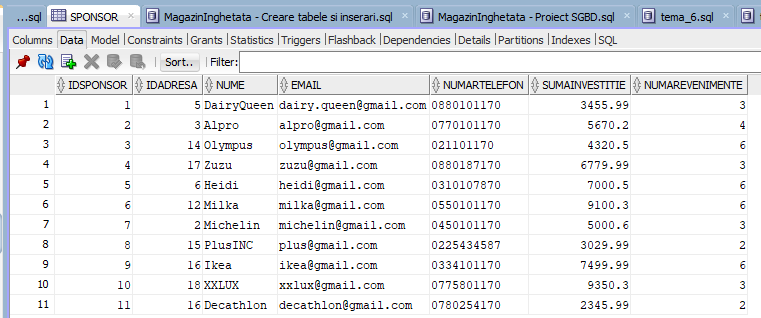


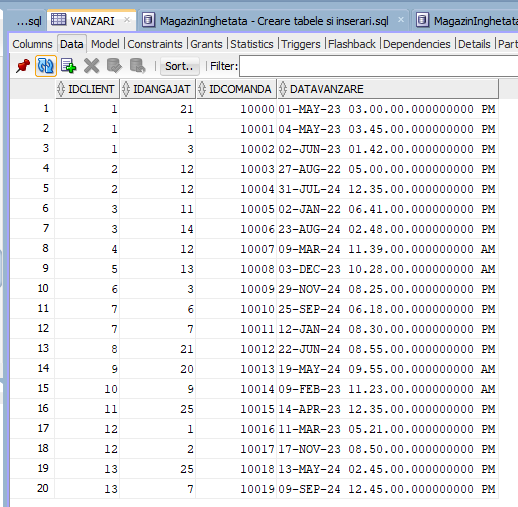
 











## Proceduri si functii PLSQL

### Exercițiul 6 (subprogram stocat independent cu 3 tipuri de colectii)

Pentru un magazin se vor afisa toti angajatii (tablou indexat) si toti clientii ai fiecarui angajat (tablou imbricat), impreuna cu cele mai recente 5 recenzii date de acel client (varray(5)).

CREATE OR REPLACE PROCEDURE Magazine\_Angajati\_si\_Recenzii(id\_magazin IN Magazine.IDMagazin%TYPE)

IS

    TYPE tablou\_indexat\_angajati IS TABLE OF Angajati.IDAngajat%TYPE INDEX BY BINARY\_INTEGER;-- Angajatii care lucreaza la magazin

    tablou\_ang tablou\_indexat\_angajati;

    nume\_ang Angajati.Nume%TYPE;

    prenume\_ang Angajati.Prenume%TYPE;

    TYPE tablou\_imbricat\_clienti IS TABLE OF Clienti.IDClient%TYPE;-- Clientii fiecarui angajat

    tablou\_clienti tablou\_imbricat\_clienti;

    TYPE varray\_recenzii IS VARRAY(5) OF Recenzii.IDRecenzie%TYPE;-- Primele 5 recenzii reprezinta de fapt un varray cu IDRecenzie

    vector\_rec varray\_recenzii := varray\_recenzii();-- Initializare varray

    rating Recenzii.Rating%TYPE;

    comm Recenzii.Comentariu%TYPE;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('Detalii angajati si clienti afisate pentru magazinul '||id\_magazin);

    SELECT IDAngajat BULK COLLECT INTO tablou\_ang

    FROM Angajati

    WHERE IDMagazin = id\_magazin;-- Toti angajatii cu FK = id\_magazin dat ca parametru

    FOR i IN 1..tablou\_ang.COUNT LOOP

        SELECT Nume, Prenume INTO nume\_ang, prenume\_ang

        FROM Angajati

        WHERE IDAngajat = tablou\_ang(i);

        DBMS\_OUTPUT.PUT\_LINE('Angajatul cu IDAngajat '||tablou\_ang(i)||' ('||nume\_ang||' '||prenume\_ang||') a vandut comenzi la clientii:');

        SELECT DISTINCT(IDClient) BULK COLLECT INTO tablou\_clienti

        -- Atentie la conditia de DISTINCT deoarece pot exista doua comenzi diferite care sa aiba valorile de IDAngajat si IDClient aceleasi, deci duplicat la IDClient

        FROM Vanzari

        WHERE IDAngajat = tablou\_ang(i);-- Clientii angajatului cu IDAngajat tablou\_ang(i)

        FOR j IN 1..tablou\_clienti.COUNT LOOP

            DBMS\_OUTPUT.PUT\_LINE('Client: '||tablou\_clienti(j));

            -- 5 recenzii recente ale clientului curent

            SELECT IDRecenzie

            BULK COLLECT INTO vector\_rec

            FROM Recenzii

            WHERE IDClient = tablou\_clienti(j)

            ORDER BY DataRecenzie DESC;

            FOR k IN 1..LEAST(vector\_rec.COUNT, vector\_rec.LIMIT) LOOP

                SELECT Rating, Comentariu INTO rating, comm

                FROM Recenzii

                where IDRecenzie = vector\_rec(k);

                -- Practic daca in vector\_rec pentru clientul curent sunt mai putin de 5 recenzii contorul merge pana la vector\_rec.COUNT

                -- Altfel, daca sunt fix 5 recenzii in vector\_rec, se merge pana la vector\_rec.LIMIT

                DBMS\_OUTPUT.PUT\_LINE('Recenzie cu IDRecenzie '||vector\_rec(k)||', rating '||rating||', comentariu '||comm);

            END LOOP;

        END LOOP;

        DBMS\_OUTPUT.PUT\_LINE('');

    END LOOP;

END Magazine\_Angajati\_si\_Recenzii;

/

BEGIN

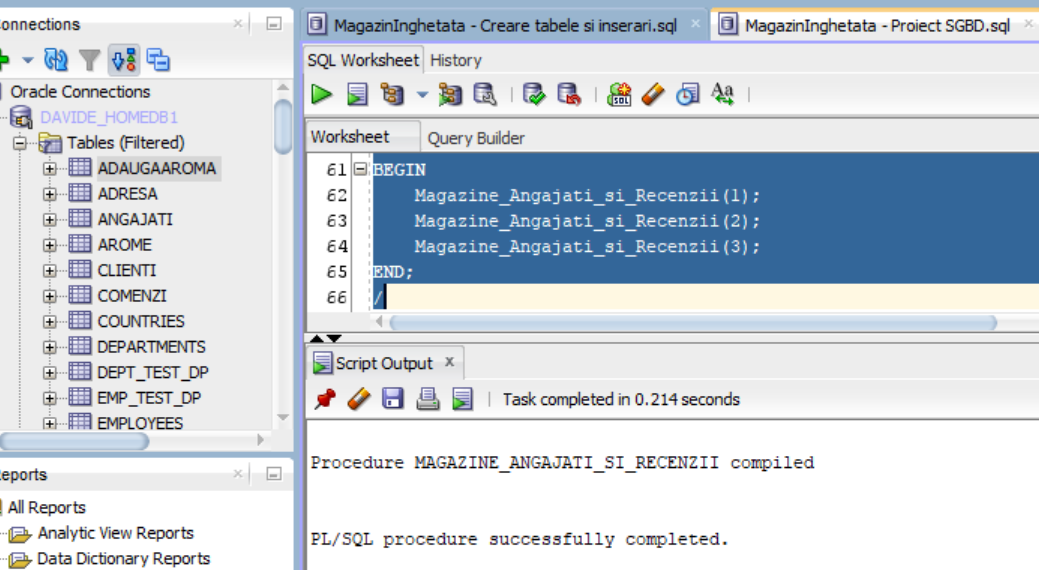
    Magazine\_Angajati\_si\_Recenzii(1);

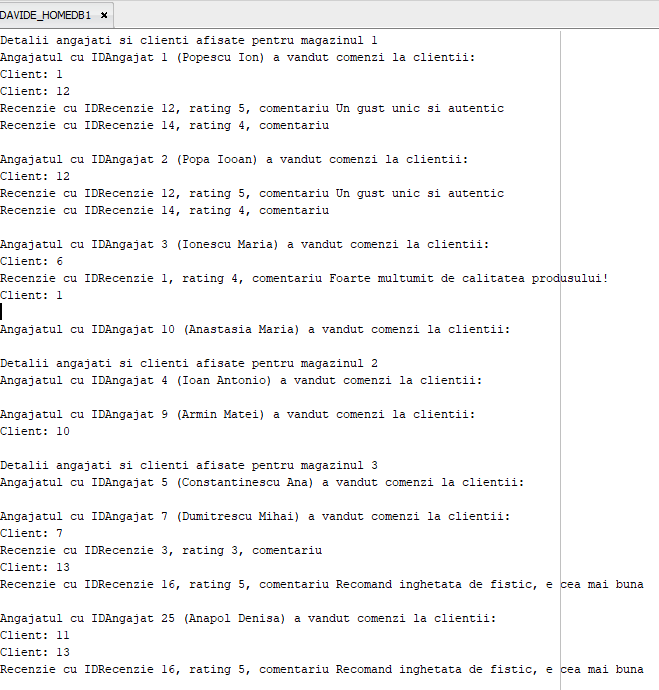
    Magazine\_Angajati\_si\_Recenzii(2);

    Magazine\_Angajati\_si\_Recenzii(3);

END;

/





### Exercițiul 7 (subprogram stocat independent 2 tipuri de cursoare, unul fiind parametrizat de celalalt cursor)

Pentru un numar k de arome, se vor prelua primele k arome ordonate descrescator dupa pret, se vor afisa toate comenzile care au suma cantitatii de arome vanduta peste media cantitatii per comanda.

2 cursoare: c\_arome, c\_comenzi(IDAroma) parametrizat

CREATE OR REPLACE PROCEDURE Afiseaza\_Arome\_Si\_Comenzi(k IN NUMBER)

IS

    CURSOR c\_arome IS

        SELECT IDAroma, Pret

        FROM Arome

        ORDER BY Pret DESC;-- Cursor simplu refeitor la detalii arome ordonate dupa pret desc

    id\_aroma Arome.IDAroma%TYPE;

    pret Arome.Pret%TYPE;

    CURSOR c\_comenzi(id IN Arome.IDAroma%TYPE) IS

        SELECT AA.IDComanda, SUM(AA.Cantitate)

        FROM AdaugaAroma AA

        WHERE AA.IDAroma = id

        GROUP BY AA.IDComanda;-- Cursor parametrizat dupa aroma pentru comenzile ce vor contine aroma aceea

    id\_comanda Vanzari.IDComanda%TYPE;

    suma\_cantitate NUMBER;

    medie\_cantitate NUMBER;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('Primele k arome in functie de pret descrescator ('||k||'):');

    OPEN c\_arome;

    FOR i IN 1..k LOOP

        FETCH c\_arome INTO id\_aroma, pret;

        EXIT WHEN c\_arome%NOTFOUND;

        DBMS\_OUTPUT.PUT\_LINE('Aroma cu IDAroma '||id\_aroma||' si pret '||pret||' ocupa pozitia '||i);

        -- Average-ul poate fi calculat inainte de a deschide cursorul c\_comenzi

        -- De retinut ca average-ul depinde de IDAroma, insemnand ca vor avea diferite valori deoarece unele arome sunt mai vandute decat celelalte

        SELECT AVG(Suma) INTO medie\_cantitate

        FROM (  SELECT SUM(AA.Cantitate) AS Suma-- Ce se afla in from e exact la fel cu ce e in cursor, e mai usor sa fac average deoarece stiu numarul de comenzi datorita group by IDComanda

                FROM AdaugaAroma AA

                WHERE AA.IDAroma = id\_aroma

                GROUP BY AA.IDComanda );

        OPEN c\_comenzi(id\_aroma);

        LOOP

            FETCH c\_comenzi INTO id\_comanda, suma\_cantitate;

            EXIT WHEN c\_comenzi%NOTFOUND;

            IF suma\_cantitate > medie\_cantitate THEN

                DBMS\_OUTPUT.PUT\_LINE('Comanda cu IDComanda '||id\_comanda||' are suma cantitatilor mai mare decat media ('||suma\_cantitate||'>'||medie\_cantitate||')');

            ELSE

                DBMS\_OUTPUT.PUT\_LINE('Comanda cu IDComanda '||id\_comanda||' NU are suma cantitatilor mai mare decat media ('||suma\_cantitate||'<='||medie\_cantitate||')');

            END IF;

        END LOOP;

        CLOSE c\_comenzi;

    END LOOP;

    CLOSE c\_arome;

    DBMS\_OUTPUT.PUT\_LINE('');

END Afiseaza\_Arome\_Si\_Comenzi;

/

BEGIN

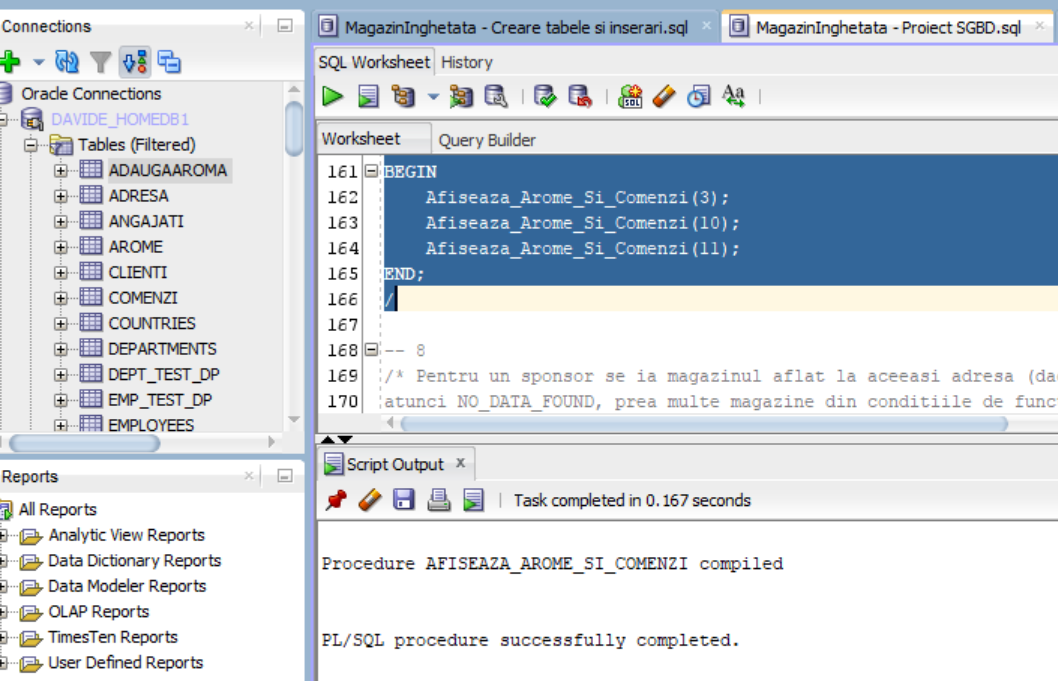
    Afiseaza\_Arome\_Si\_Comenzi(3);

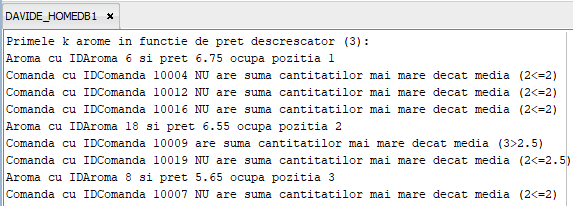
    Afiseaza\_Arome\_Si\_Comenzi(10);

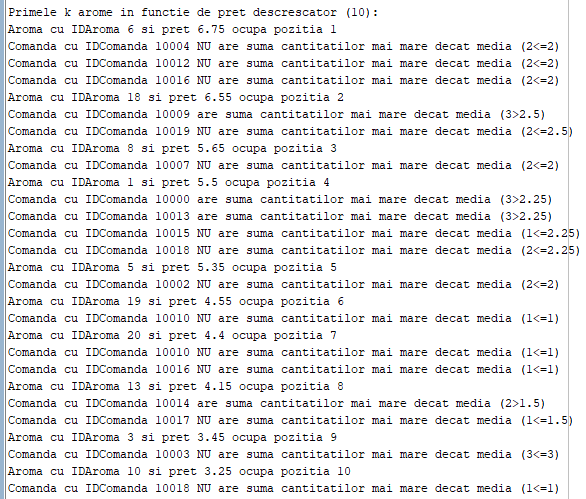
    Afiseaza\_Arome\_Si\_Comenzi(11);

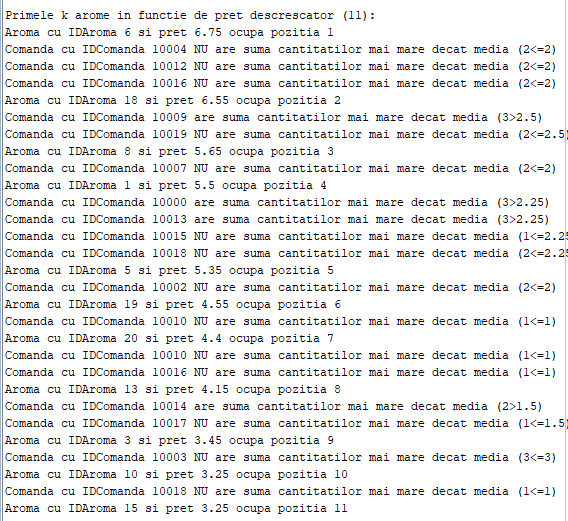
END;

/









### Exercițiul 8 (Functie exceptii no\_data\_found, Too\_Many\_rows, JOIN 3 tabele)

Pentru un sponsor se ia magazinul aflat la aceeasi adresa (daca nu exista niciun magazin la IDAresa FK din Sponsor, atunci NO\_DATA\_FOUND) si se afiseaza angajatul cu numarul maxim de arome distincte procesate (NO\_DATA\_FOUND daca nu exista niciun angajat care sa lucreze la magazinul gasit, TOO\_MANY\_ROWS daca exista 2 sau mai multi angajati cu numarul maxim de arome procesate, deoarece am stabilit ca vreau doar un nume de angajat).

-- De retinut ca magazinele cu id de la 1 la 13 au aceeasi adresa ca si id-ul, iar restul adreselor de la 14-18 pot apartine doar sponsorilor

-- Deci adresele in comun de la magazine si sponsori sunt intre 1-13, dar trebuie verificat in tabela Sponsor

CREATE OR REPLACE FUNCTION Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(id\_sponsor IN Sponsor.IDSponsor%TYPE)

RETURN VARCHAR

IS

    id\_magazin Magazine.IDMagazin%TYPE;

    CURSOR c\_angajat(id Magazine.IDMagazin%TYPE) IS

        SELECT A.Nume, COUNT(DISTINCT(AA.IDAroma)) AS nr\_arome\_distincte

        FROM Angajati A

        JOIN Vanzari V ON A.IDAngajat = V.IDAngajat

        JOIN Comenzi C ON V.IDComanda = C.IDComanda

        JOIN AdaugaAroma AA ON AA.IDComanda = C.IDComanda

        WHERE A.IDMagazin = id

        GROUP BY A.Nume;

    TYPE record\_ang IS RECORD (

        nume\_ang Angajati.Nume%TYPE,

        nr\_arome\_distincte NUMBER

    );

    TYPE tablou\_indexat\_angajat IS TABLE OF record\_ang INDEX BY BINARY\_INTEGER;

    tablou\_ang tablou\_indexat\_angajat;

    nume\_ang\_cautat VARCHAR(50);

    maxim\_arome\_distincte NUMBER := 0;-- Maxim de arome la nivel de magazin

    count\_ang\_cu\_nr\_maxim\_arome NUMBER := 0;-- Numara angajatii cu numarul maxim de arome procesate la magazinul respectiv

BEGIN

    -- Un alt bloc begin poentru a trata NO\_DATA\_FOUND de la magazine, din moment ce va exista si la angajati

    BEGIN

        SELECT M.IDMagazin

        INTO id\_magazin

        FROM Magazine M

        JOIN Sponsor S ON (S.IDAdresa = M.IDAdresa)-- Atentie! JOIN in functie de adresa si nu IDSponsor

        WHERE S.IDSponsor = id\_sponsor;

        IF id\_magazin IS NULL THEN-- Atunci nu exista niciun magazin asociat cu adresa sponsorului

            RAISE NO\_DATA\_FOUND;

        END IF;

    EXCEPTION

        WHEN NO\_DATA\_FOUND THEN

            RETURN 'Nu exista niciun magazin asociat cu adresa sponsorului IDSponsor '||id\_sponsor;

    END;

    BEGIN

        OPEN c\_angajat(id\_magazin);-- Numele angajatilor si numar de arome distincte procesate de ei pentru magazinul gasit anterior (cursor parametrizat)

        FETCH c\_angajat BULK COLLECT INTO tablou\_ang;

        CLOSE c\_angajat;

        IF tablou\_ang.COUNT = 0 THEN-- Daca BULK COLLECT intoarce un tablou gol, inseamna ca nu exista niciun angajat care lucreaza la magazin

            RAISE NO\_DATA\_FOUND;

        END IF;

        -- Parcurg toti angajatii pentru a-l gasi pe cel cu numar maxim de arome distincte

        -- Nu il gasesc chiar in acest moment, deoarece stiu ca pot exista mai multi, asa ca retin in variabila count\_ang\_cu\_nr\_maxim\_arome nr de ang cu proprietatea asta

        FOR i IN 1..tablou\_ang.COUNT LOOP

            IF tablou\_ang(i).nr\_arome\_distincte > maxim\_arome\_distincte THEN

                maxim\_arome\_distincte := tablou\_ang(i).nr\_arome\_distincte;

                count\_ang\_cu\_nr\_maxim\_arome := 1;-- Un nou maxim inseamna ca e un singur angajat cu acest maxim si se reseteaza counter-ul

            ELSIF tablou\_ang(i).nr\_arome\_distincte = maxim\_arome\_distincte THEN

                count\_ang\_cu\_nr\_maxim\_arome := count\_ang\_cu\_nr\_maxim\_arome + 1;-- Mai multi angajati cu acelasi maxim

            END IF;

        END LOOP;

        IF count\_ang\_cu\_nr\_maxim\_arome > 1 THEN-- Prea multi ang

            RAISE TOO\_MANY\_ROWS;

        END IF;

        -- M-am asigurat ca e doar unul, ii gasesc numele si termin

        SELECT A.Nume INTO nume\_ang\_cautat

        FROM Angajati A

        WHERE A.IDAngajat = (   SELECT A2.IDAngajat -- Subcererea va returna neaparat un IDAngajat, din moment ce am stabilit WHERE A.IDAngajat =

                                FROM Angajati A2

                                JOIN Vanzari V ON (A2.IDAngajat = V.IDAngajat)

                                JOIN Comenzi C ON (V.IDComanda = C.IDComanda)

                                JOIN AdaugaAroma AA ON (AA.IDComanda = C.IDComanda)

                                WHERE A2.IDMagazin = id\_magazin

                                GROUP BY A2.IDAngajat

                                HAVING COUNT(DISTINCT AA.IDAroma) = maxim\_arome\_distincte );

        return nume\_ang\_cautat;

        -- Asa ar fi fost daca era gandita ca o procedura (sa afiseze mesaje)

        --DBMS\_OUTPUT.PUT\_LINE('Angajatul cu cele mai multe arome distincte este '||nume\_ang\_cautat);

    EXCEPTION

        WHEN NO\_DATA\_FOUND THEN

            RETURN 'Nu exista angajati in magazinul cu IDMagazin '||id\_magazin;

        WHEN TOO\_MANY\_ROWS THEN

            RETURN 'Prea multi angajati cu acelasi numar de arome distincte procesate in magazinul cu IDMagazin '||id\_magazin;

    END;

END Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome;

/

BEGIN

    DBMS\_OUTPUT.PUT\_LINE(Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(1));-- NO\_DATA\_FOUND angajati care lucreaza

    DBMS\_OUTPUT.PUT\_LINE(Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(7));

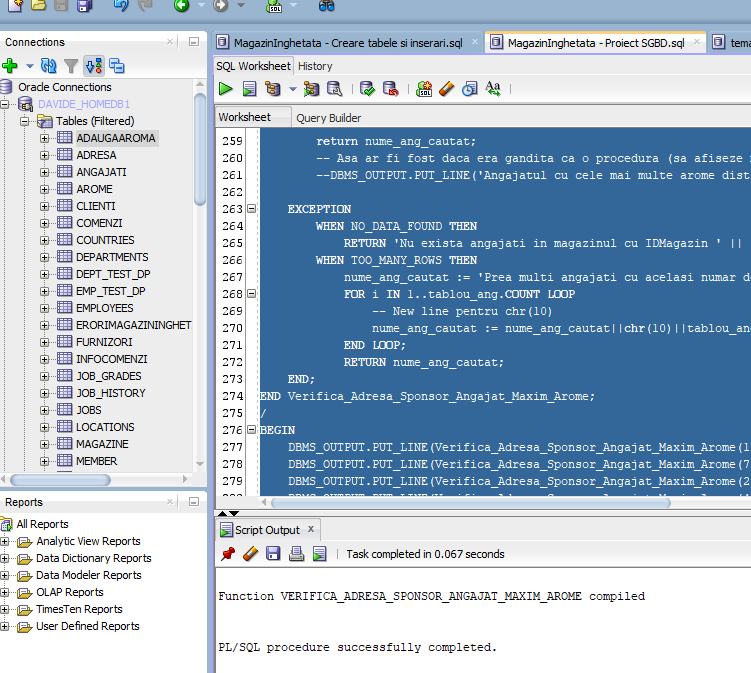
    DBMS\_OUTPUT.PUT\_LINE(Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(2));

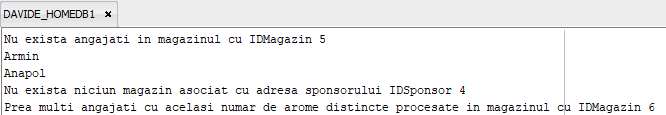
    DBMS\_OUTPUT.PUT\_LINE(Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(4));-- NO\_DATA\_FOUND magazine la adresa

    DBMS\_OUTPUT.PUT\_LINE(Verifica\_Adresa\_Sponsor\_Angajat\_Maxim\_Arome(5));-- TOO\_MANY\_ROWS angajati cu acelasi max de arome distincte

END;

/





### Exercițiul 9 (PROCEDURA CU 2 PARAMETRI, 2 EXCEPTII PROPRII, JOIN 5 TABELE)

Pentru un client se vor afisa recenziile postate (NO\_REVIEW\_YET daca nu a postat inca niciun review) si pentru acelasi client si o adresa se vor afisa detaliile comenzilor clientului respectiv plasate de catre un angajat care lucreaza magazinul cu adresa data (NO\_ADDRESS\_FOUND).

CREATE OR REPLACE PROCEDURE Verifica\_Client\_Recenzii\_Adresa\_Comenzi(id\_client Clienti.IDClient%TYPE, id\_adresa IN Adresa.IDAdresa%TYPE)

IS

    CURSOR c\_recenzii IS-- Cursor simplu recenzii in functie de client

        SELECT Rating, Comentariu

        FROM Recenzii

        WHERE IDClient = id\_client;

    TYPE record\_recenzie IS RECORD (

        rating Recenzii.Rating%TYPE,

        comm Recenzii.Comentariu%TYPE

    );

    TYPE tablou\_indexat\_recenzii IS TABLE OF record\_recenzie INDEX BY BINARY\_INTEGER;

    tablou\_rec tablou\_indexat\_recenzii;-- Toate rezultatele cursorului vor fi puse BULK COLLECT INTO in acest tablou de recenzii

    CURSOR c\_comenzi IS

        SELECT V.IDClient, C.Prenume, V.IDComanda, V.IDAngajat, A.Nume, V.DataVanzare, M.IDMagazin, M.IDAdresa, AD.Oras, AD.Strada

        FROM Clienti C

        JOIN Vanzari V ON (C.IDClient = V.IDClient)

        JOIN Angajati A ON (V.IDAngajat = A.IDAngajat)

        JOIN Magazine M ON (A.IDMagazin = M.IDMagazin)

        jOIN Adresa AD ON(AD.IDAdresa = M.IDAdresa)

        WHERE M.IDAdresa = id\_adresa AND V.IDClient = id\_client;-- Multe detalii legate de vanzare, join 5 tabele astefl incat vanzarea are clientul dat, iar magazinul in care lucreaza ang ce a prelucrat comanda e la adresa data

    TYPE record\_vanzare IS RECORD (

        id\_client Clienti.IDClient%TYPE,

        prenume\_client Clienti.Prenume%TYPE,

        id\_comanda Comenzi.IDComanda%TYPE,

        id\_ang Angajati.IDAngajat%TYPE,

        nume\_ang Angajati.Nume%TYPE,

        data\_vanzare Vanzari.DataVanzare%TYPE,

        id\_magazin\_ang Magazine.IDMagazin%TYPE,

        id\_adresa Adresa.IDAdresa%TYPE,

        oras Adresa.Oras%TYPE,

        strada Adresa.Strada%TYPE

    );

    TYPE tablou\_indexat\_vanzari IS TABLE OF record\_vanzare INDEX BY BINARY\_INTEGER;

    tablou\_vanzari tablou\_indexat\_vanzari;-- Inca o data BULK COLLECT INTO

    NO\_REVIEW\_YET EXCEPTION;

    NO\_ADDRESS\_FOUND EXCEPTION;

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('Clientul '||id\_client||':');

    OPEN c\_recenzii;

    FETCH c\_recenzii BULK COLLECT INTO tablou\_rec;

    CLOSE c\_recenzii;

    IF tablou\_rec.COUNT = 0 THEN-- Nicio recenzie nu a fost gasita, deci NO\_REVIEW\_YET asociat clientului dat

        RAISE NO\_REVIEW\_YET;

    END IF;

    FOR i IN 1..tablou\_rec.COUNT LOOP

        IF i = 1 THEN

            DBMS\_OUTPUT.PUT\_LINE('Primul comentariu:');

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('Al '||i||'-lea comentariu:');

        END IF;

        DBMS\_OUTPUT.PUT\_LINE('rating '||tablou\_rec(i).rating||', comentariu '||tablou\_rec(i).comm);

    END LOOP;

    OPEN c\_comenzi;

    FETCH c\_comenzi BULK COLLECT INTO tablou\_vanzari;

    CLOSE c\_comenzi;

    IF tablou\_vanzari.COUNT = 0 THEN-- Nu merge sa gaseasca adresa magazinului (sigur orice client are cel putin o vanzare deci nu e un caz pe care ar trebui sa-l tratez)

        RAISE NO\_ADDRESS\_FOUND;

    END IF;

    FOR i IN 1..tablou\_vanzari.COUNT LOOP

        DBMS\_OUTPUT.PUT\_LINE('Clientul cu prenumele '||tablou\_vanzari(i).prenume\_client||' a plasat comanda '||tablou\_vanzari(i).id\_comanda||' procesata de angajatul '||tablou\_vanzari(i).id\_ang||' cu numele '||tablou\_vanzari(i).nume\_ang);

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

        DBMS\_OUTPUT.PUT\_LINE('data vanzare: '||TO\_CHAR(tablou\_vanzari(i).data\_vanzare, 'DD-MON-YYYY')||', magazinul la care lucreaza angajatul: '||tablou\_vanzari(i).id\_magazin\_ang||', adresa magazinului: '||tablou\_vanzari(i).oras||', '||tablou\_vanzari(i).strada);

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('');

EXCEPTION

    WHEN NO\_REVIEW\_YET THEN

        DBMS\_OUTPUT.PUT\_LINE('Clientul '||id\_client||' nu a postat nicio recenzie inca');

        DBMS\_OUTPUT.PUT\_LINE('');-- Pentru afisare cu new line

    WHEN NO\_ADDRESS\_FOUND THEN

        DBMS\_OUTPUT.PUT\_LINE('Nicio comanda nu a fost identificata ca fiind procesata de un angajat care lucreaza la magazinul cu adresa '||id\_adresa);

        DBMS\_OUTPUT.PUT\_LINE('');

END Verifica\_Client\_Recenzii\_Adresa\_Comenzi;

/

BEGIN

    Verifica\_Client\_Recenzii\_Adresa\_Comenzi(1, 2);-- NO\_REVIEW\_YET

    Verifica\_Client\_Recenzii\_Adresa\_Comenzi(3, 6);

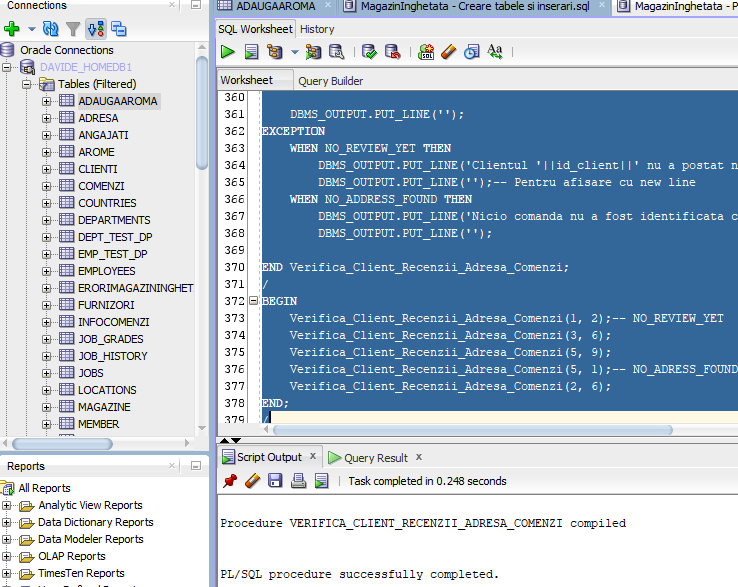
    Verifica\_Client\_Recenzii\_Adresa\_Comenzi(5, 9);

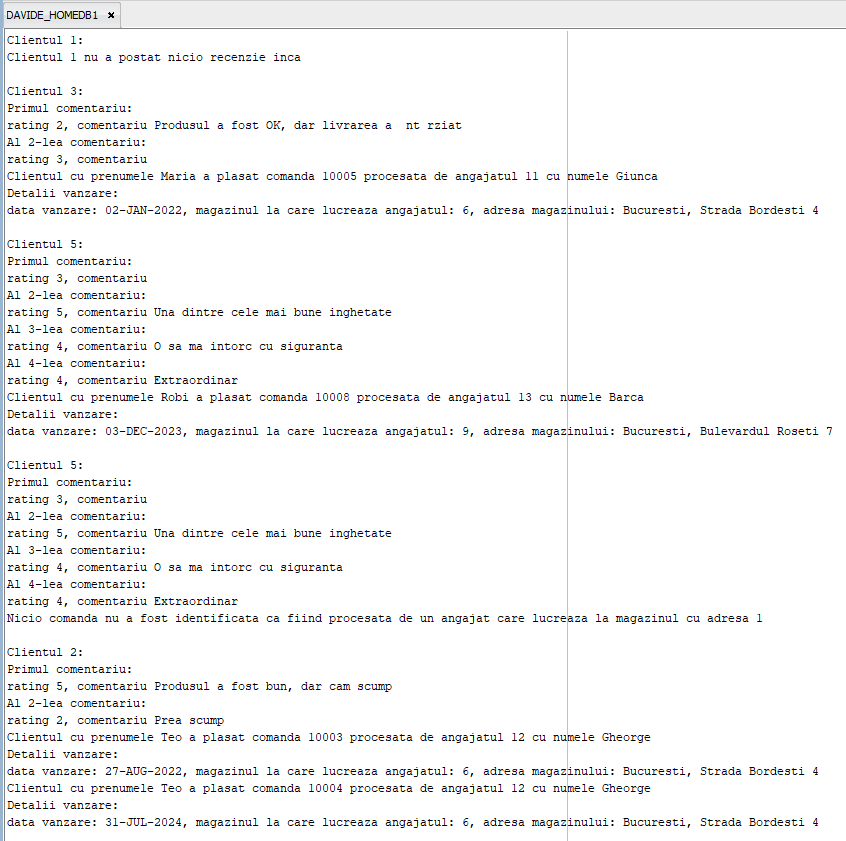
    Verifica\_Client\_Recenzii\_Adresa\_Comenzi(5, 1);-- NO\_ADRESS\_FOUND

    Verifica\_Client\_Recenzii\_Adresa\_Comenzi(2, 6);

END;

/





## TRIGGERS

### Exercițiul 10 (LMD la nivel de comanda)

Inserarea in tabela Magazine genereaza modificari a rentabilitatii la ora a magazinului aflat la aceeasi adresa cu sponsorul: ma folosesc de o noua tabela care grupeaza ce se afla in select dupa IDSponsor si cu ajutorul unui cursor care e parcurs la fiecare inserare, modifica datele din tabela Rentabilitate\_Magazine\_Bazata\_Sponsor in fucntie de IDSponsor FK din noua inserare in Magazine. Legat de regulile de modificare a rentabilitatii la ora: RentabilitateOra creste cu 20% pentru suma totala de investitie e mai mare ca 10000, altfel scade cu 10%.

--DROP TABLE Rentabilitate\_Magazine\_Bazata\_Sponsor;

-- Vreau ca noul tabel sa fie grupat dupa IDMagazin astfel sa pot sa gasesc numarul de magazine sustinute de acelasi sponsor si o medie a rentabilitatii la ora dintre aceste magazine sustinute

CREATE TABLE Rentabilitate\_Magazine\_Bazata\_Sponsor AS

SELECT S.IDSponsor, COUNT(M.IDMagazin) AS NrMagazine, AVG(M.RentabilitateOra) AS RentabilitateOra

FROM Magazine M

JOIN Sponsor S ON (M.IDSponsor = S.IDSponsor)

GROUP BY S.IDSponsor;

-- Suma de sponsorizare e 0 in momentul asta

ALTER TABLE Rentabilitate\_Magazine\_Bazata\_Sponsor

ADD SumaSponsorizare NUMBER DEFAULT 0;

CREATE OR REPLACE TRIGGER Trigger\_Sponsor\_Rentabilitate\_Magazine-- Trigger la nivel de comanda (fara for each row)

AFTER INSERT ON Magazine

DECLARE

    CURSOR c\_sponsori\_info IS

        SELECT S.IDSponsor, COUNT(M.IDMagazin), AVG(M.RentabilitateOra), SUM(S.SumaInvestitie)

        FROM Magazine M

        JOIN Sponsor S ON (M.IDSponsor = S.IDSponsor)

        GROUP BY S.IDSponsor;-- Cursorul practic imi recalculeaza numarul de magazine, media si totalul sumei investite de sponsor

    id\_sponsor Sponsor.IDSponsor%TYPE;

    nr\_magazine NUMBER;

    medie\_rentabilitate\_ora Magazine.RentabilitateOra%TYPE;

    suma\_sponsorizare\_total NUMBER;

BEGIN

    OPEN c\_sponsori\_info;

    LOOP

        FETCH c\_sponsori\_info INTO id\_sponsor, nr\_magazine, medie\_rentabilitate\_ora, suma\_sponsorizare\_total;

        EXIT WHEN c\_sponsori\_info%NOTFOUND;

        UPDATE Rentabilitate\_Magazine\_Bazata\_Sponsor-- dau update la coloanele din tabel cu ce se afla curent in cursor

        SET NrMagazine = nr\_magazine,

            SumaSponsorizare = suma\_sponsorizare\_total,

            RentabilitateOra =

            CASE

                WHEN suma\_sponsorizare\_total >= 10000 THEN LEAST(medie\_rentabilitate\_ora + (medie\_rentabilitate\_ora \* 0.2), 100)

                ELSE GREATEST(medie\_rentabilitate\_ora - (medie\_rentabilitate\_ora \* 0.1), 0)

            END-- Daca suma totala investita e mai mare ca 10000, creste rentabilitatea la ora, dar exprimat in functie de medie\_rentabilitate\_ora ce e intotdeauna recalculat in cursor

        WHERE IDSponsor = id\_sponsor;

        -- Cu LEAST si GREATEST ma aisgur ca nu intrec limitele logice (chiar daca nu am pus un constraint in noul tabel)

    END LOOP;

    CLOSE c\_sponsori\_info;

END Trigger\_Sponsor\_Rentabilitate\_Magazine;

/

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

VALUES (14, 8, 2, 11, 'Pufic 14', 10);-- Incrementez magazinele sponsorului 11, rentabilitate = 41.517, suma = 7037.97

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

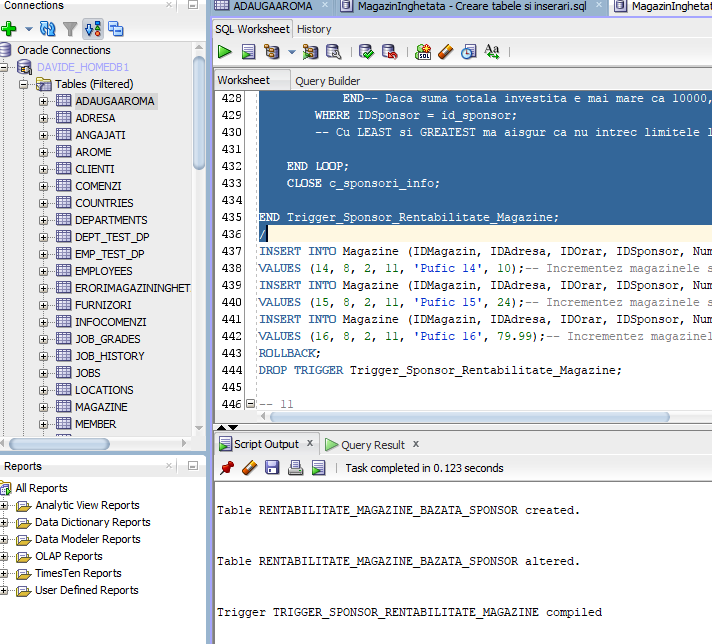
VALUES (15, 8, 2, 11, 'Pufic 15', 24);-- Incrementez magazinele sponsorului 11, scade de la 41.517 la 36.54 rentabilitatea si creste suma investita cu inca 2345.99 (2345.99 + 7037.97 = 9383.96)

INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

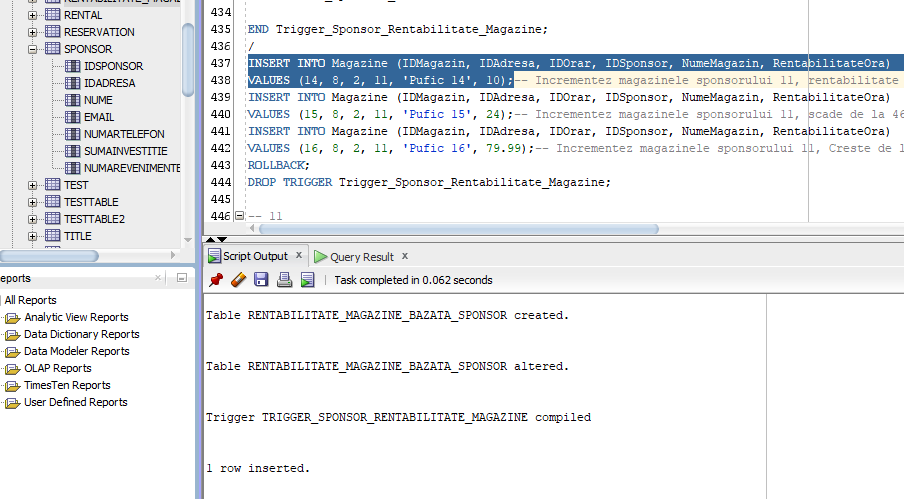
VALUES (16, 8, 2, 11, 'Pufic 16', 79.99);-- Incrementez magazinele sponsorului 11, Creste de la 36.54 la 58.176 rentabilitatea si creste suma investita la 11729.95

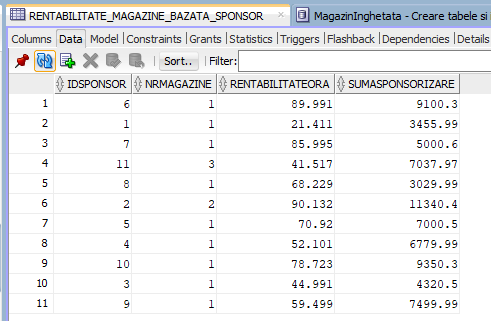
ROLLBACK;

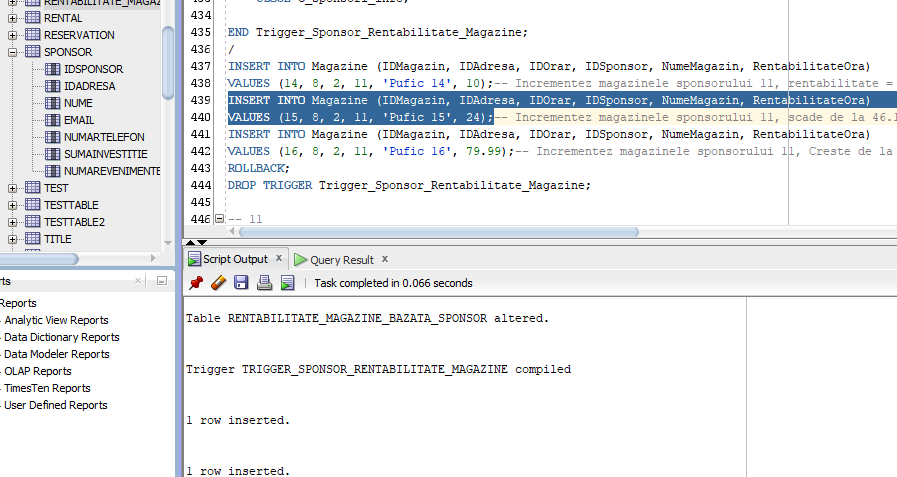
DROP TRIGGER Trigger\_Sponsor\_Rentabilitate\_Magazine;

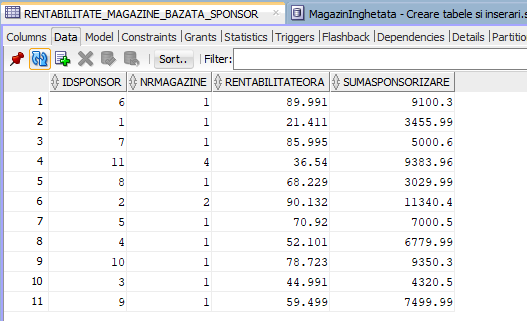




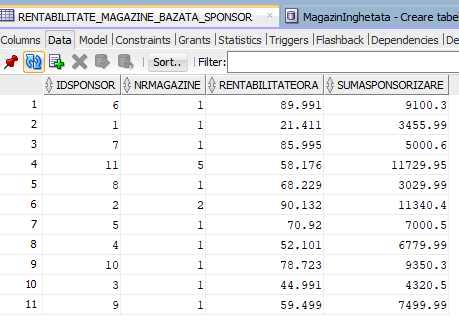


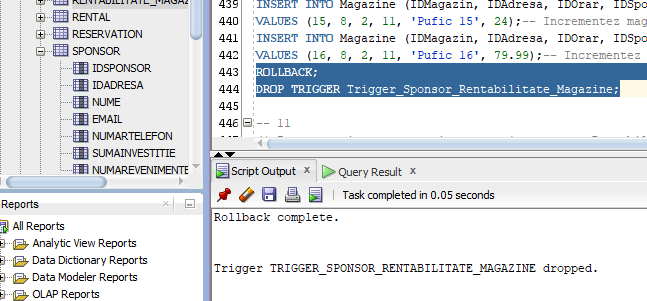












### Exercițiul 11 (LMD la nivel de linie)

Pentru un inserarea unui nou angajat, creste RentabilitateOra cu 8% pentru toate magazinele cu acelasi numar de angajati daca magazinul la care lucreaza angajatul inainte sa fie adaugat are un numar de angajati mai mic decat 4 si egal, sau creste cu 2% daca numarul de angajati e mai mare. De retinut ca numarul de angajati e calculat fara angajatul care urmeaza sa fie adaugat. Presupun ca doar peste o valoare de 4 angajati se genereaza mai putin estimated revenue pentru acel magazin, adica un procent mai mic de RentabilitateOra se adauga per angajat daca e intrecuta limita.

CREATE OR REPLACE TRIGGER Trigger\_Rentabilitate\_Angajati

BEFORE INSERT ON Angajati-- Dupa inserare

FOR EACH ROW-- Se parcurge fiecare rand din Magazine (desi se modifica un singur magazin) si se modifica dupa inserarea unui angajat, dar e nevoie de FOR EACH ROW pentru operatorul :NEW

DECLARE

    id\_magazin Magazine.IDMagazin%TYPE;

    nr\_curent\_ang NUMBER;

BEGIN

    id\_magazin := :NEW.IDMagazin;

    SELECT COUNT(\*)

    INTO nr\_curent\_ang

    FROM Angajati

    WHERE IDMagazin = id\_magazin;-- Gasesc numarul de angajati inainte sa inserez noul ang

    -- Rentabilitatea creste cu mai mult daca la un magazin sunt mai putini angajati, cand incep sa fie mai multi, cresterea nu mai e la fel de mare

    IF nr\_curent\_ang <= 4 THEN

        UPDATE Magazine

        SET RentabilitateOra = LEAST(RentabilitateOra + RentabilitateOra \* 0.08, 99)

        WHERE IDMagazin IN (    SELECT IDMagazin

                                FROM Angajati

                                GROUP BY IDMagazin

                                HAVING COUNT(\*) = nr\_curent\_ang );-- Are sens trigger cu for each row pentru ca se actualizeaza mai multe magazine cu acelasi numar de angajati ca cel la care tocmai s-a facut insert de FK in Angajati

    ELSE

        UPDATE Magazine

        SET RentabilitateOra = LEAST(RentabilitateOra + RentabilitateOra \* 0.02, 99)

        WHERE IDMagazin IN (    SELECT IDMagazin

                                FROM Angajati

                                GROUP BY IDMagazin

                                HAVING COUNT(\*) = nr\_curent\_ang );

    END IF;

END Trigger\_Rentabilitate\_Angajati;

/

-- Magazinul cu IDMagazin 11 are 3 angajati si rentabilitate 81.23

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (27, 11, 'Vlad', 'Mircea', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);-- 3 ang (fara cel curent) la magazinul 11, creste cu 8%: 87.73 la id\_magazin 11, 53.99 la id\_magazin 3, 99 la id\_magazin 9; creste cu 2%: 99 la id\_magazin 6

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (28, 11, 'Filipescu', 'Daniel', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);-- 4 ang la magazinul 11, creste cu 8%: 94.75 la id\_magazin 11, 76.57 la id\_magazin 1

INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

VALUES (29, 11, 'Dumitrescu', 'Dinu', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);-- 5 ang la magazinul 11, creste cu 2%: 96.65 la id\_magazin 11

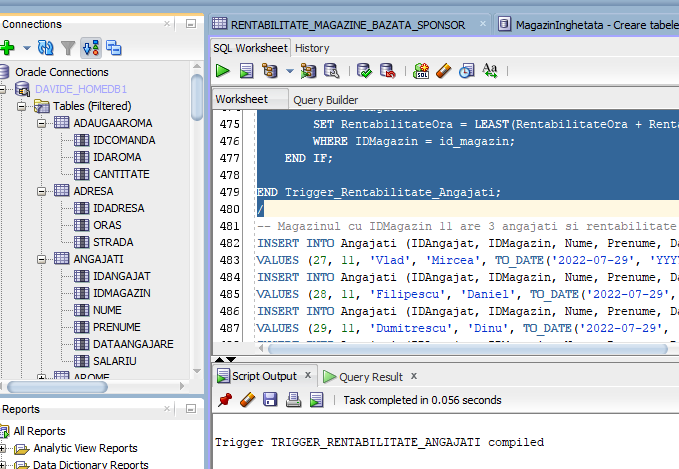
INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

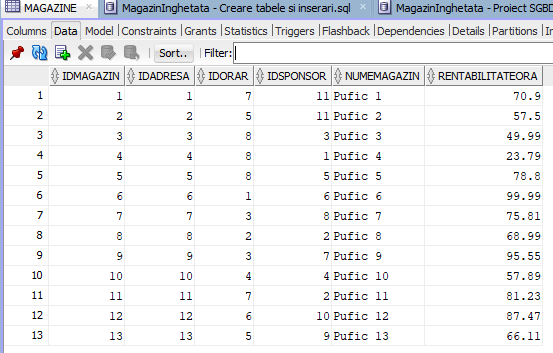
VALUES (30, 11, 'Moraru', 'Rodica', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);-- 6 ang la magazinul 11, creste cu 2%: 98.58 la id\_magazin 11

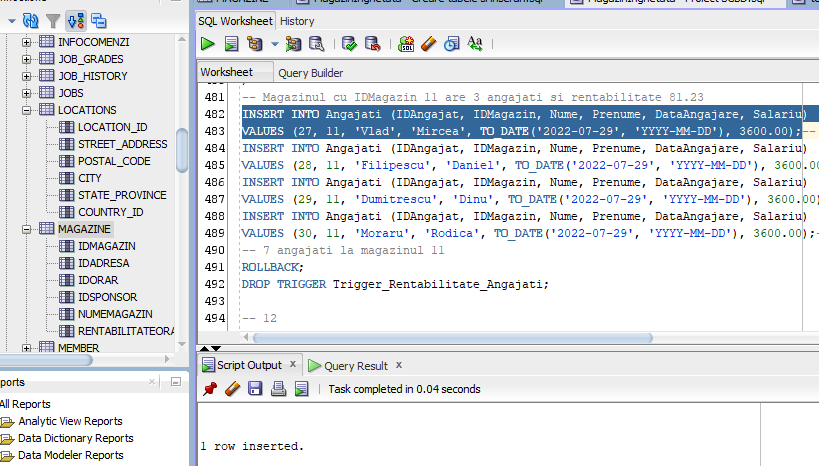
-- 7 angajati la magazinul 11

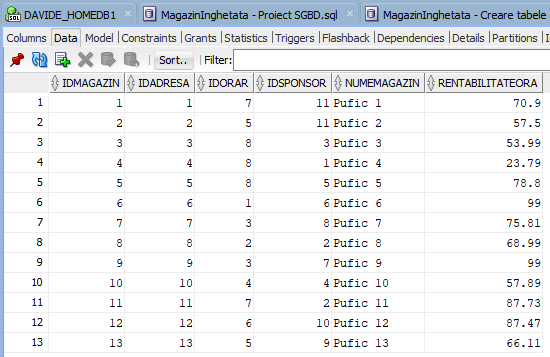
ROLLBACK;

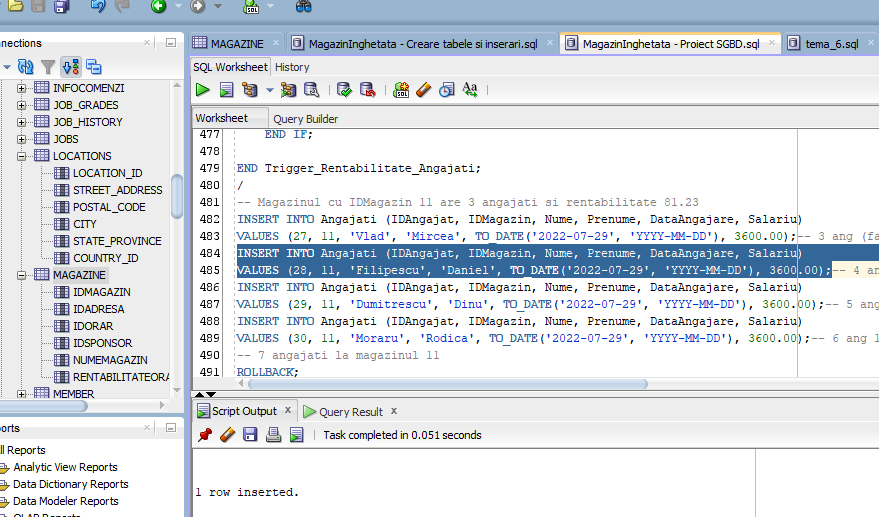
DROP TRIGGER Trigger\_Rentabilitate\_Angajati;

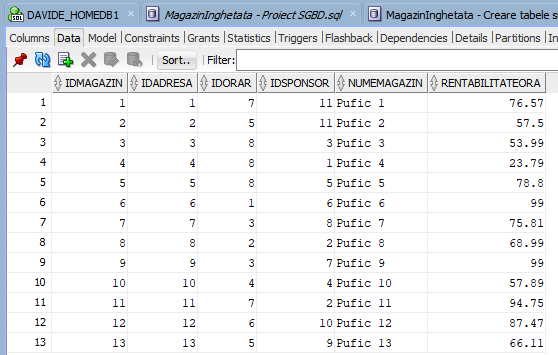


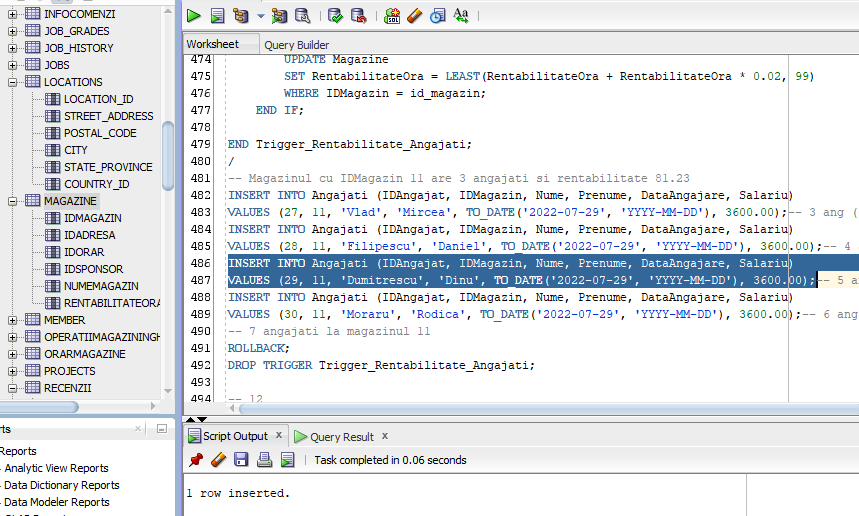


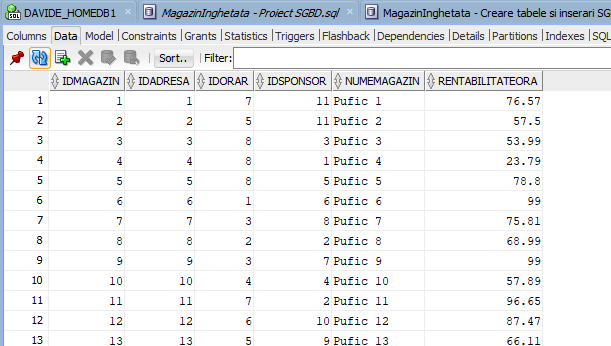


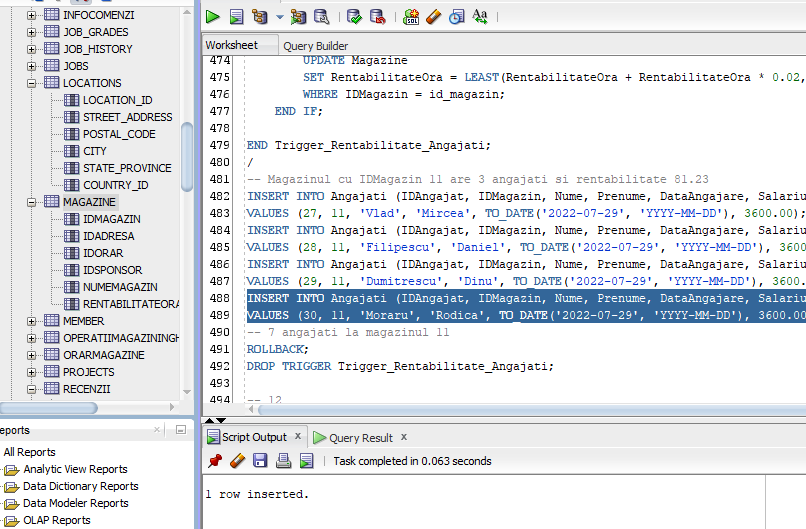


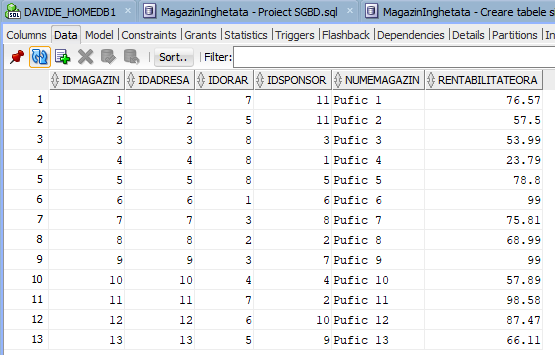


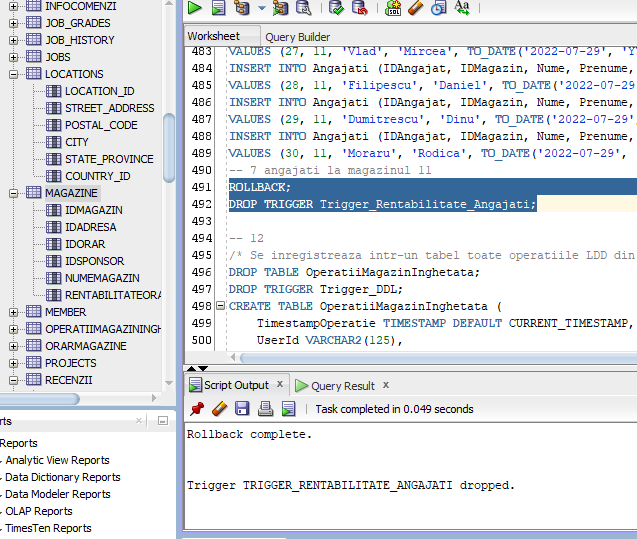












### Exercițiul 12 (LDD)

Se inregistreaza intr-un tabel toate operatiile LDD din baza de date, impreuna cu timestamp-ul la care a avut loc create/ alter/ drop.

--DROP TRIGGER Trigger\_DDL;

--DROP TABLE OperatiiMagazinInghetata;

--DROP TABLE TestTable;

CREATE TABLE OperatiiMagazinInghetata (

    TimestampOperatie TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    UserId VARCHAR2(125),

    TipEveniment VARCHAR2(50),

    NumeObiect VARCHAR2(255)-- Pe ce obiect se face actiunea

);

CREATE OR REPLACE TRIGGER Trigger\_DDL

AFTER CREATE OR ALTER OR DROP ON SCHEMA

BEGIN

    INSERT INTO OperatiiMagazinInghetata (UserId, TipEveniment, NumeObiect)

    VALUES (SYS.LOGIN\_USER, SYS.SYSEVENT, SYS.DICTIONARY\_OBJ\_NAME);

END Trigger\_DDL;

/

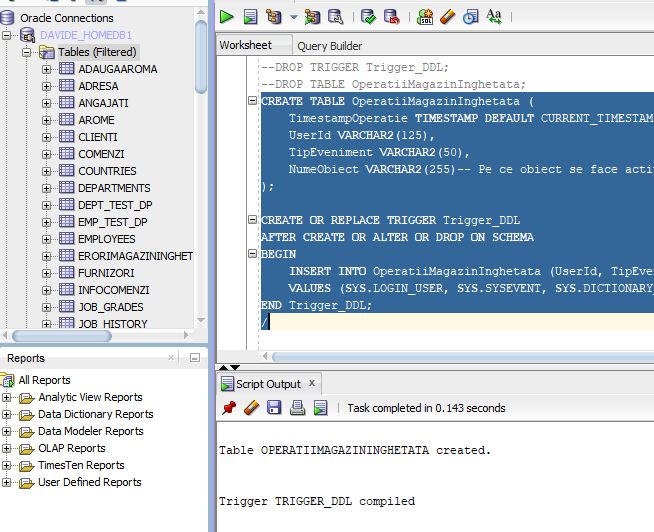
CREATE TABLE TestTable (

    ID INT PRIMARY KEY,

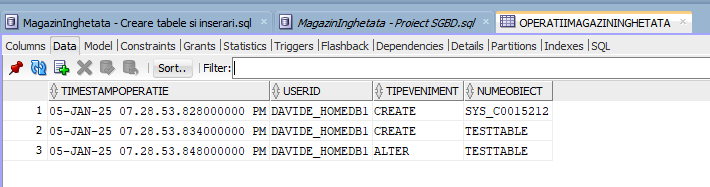
    Nume VARCHAR2(100)

);

ALTER TABLE TestTable DROP COLUMN Nume;







## Pachete

### Exercițiul 13 (2 Tipuri complexe de date, 2 functii, 2 proceduri) - Incomplet

Am un pachet care are o procedura si o functie (1 variabila globala – id\_sponsor):

* functie care primeste un angajat si verifica daca magazinul la care lucreaza e sponsorizat (true sau false)
* procedura care creaza un tablou indexat cu toti angajatii si pentru angajatii la care da true daca apelez functia anterioara, verific daca la nivelul magazinului o treime din suma salariilor angajatilor e suportata de SumaInvestitie a sponsorului

CREATE OR REPLACE PACKAGE Pachet\_Angajati AS

    id\_sponsor Sponsor.IDSponsor%TYPE;-- Variabila deoarece doresc daca la apelul functiei pentru un angajat curent sa stiu si sponsorul global (il gasesc in functie si il afisez abia in procedura)

    FUNCTION Verifica\_Sponsor\_Magazin(id\_ang IN Angajati.IDAngajat%TYPE) RETURN BOOLEAN;-- Se face apelarea functiei in interiorul procedurii

    PROCEDURE Afisare\_Angajati\_Cheltuieli\_Acoperite\_Magazin;-- Procedura nu va avea nevoie de parametrii deoarece am stabilit ca va lua toti angajatii din BD

END Pachet\_Angajati;

/

CREATE OR REPLACE PACKAGE BODY Pachet\_Angajati AS

    FUNCTION Verifica\_Sponsor\_Magazin(id\_ang Angajati.IDAngajat%TYPE)

    RETURN BOOLEAN

    IS

        bool\_sponsorizat BOOLEAN := FALSE;

    BEGIN

        SELECT M.IDSponsor INTO id\_sponsor-- Folosesc variabila globala

        FROM Magazine M

        JOIN Angajati A ON (A.IDMagazin = M.IDMagazin)

        WHERE A.IDAngajat = id\_ang;-- Primit ca parametru in functie

        IF id\_sponsor IS NOT NULL THEN

            bool\_sponsorizat := TRUE;

        END IF;

        RETURN bool\_sponsorizat;

    END Verifica\_Sponsor\_Magazin;

    PROCEDURE Afisare\_Angajati\_Cheltuieli\_Acoperite\_Magazin

    IS

        CURSOR c\_angajati IS

            SELECT A.IDAngajat, A.IDMagazin, A.Nume, A.Salariu

            FROM Angajati A;

        id\_angajat Angajati.IDAngajat%TYPE;

        id\_magazin Magazine.IDMagazin%TYPE;

        nume\_ang Angajati.Nume%TYPE;

        salariu\_ang Angajati.Salariu%TYPE;

        suma\_salarii NUMBER;-- Fac suma cu un select ca sa nu ma complic

        total\_suma\_investitie NUMBER;

    BEGIN

        OPEN c\_angajati;

        LOOP

            FETCH c\_angajati INTO id\_angajat, id\_magazin, nume\_ang, salariu\_ang;

            EXIT WHEN c\_angajati%NOTFOUND;

            DBMS\_OUTPUT.PUT\_LINE('Angajatul cu IDAngajat '||id\_angajat||', nume '||nume\_ang||', salariu '||salariu\_ang);

            IF Verifica\_Sponsor\_Magazin(id\_angajat) THEN

                DBMS\_OUTPUT.PUT\_LINE('Angajatul '||id\_angajat||' lucreaza la un magazin sponsorizat cu IDMagazin '||id\_magazin);

                DBMS\_OUTPUT.PUT\_LINE('Magazinul are sponsorul '||id\_sponsor);

                SELECT SUM(A.Salariu) INTO suma\_salarii

                FROM Angajati A

                WHERE A.IDMagazin = id\_magazin;

                SELECT SUM(S.SumaInvestitie) INTO total\_suma\_investitie

                FROM Magazine M

                JOIN Sponsor S ON (M.IDSponsor = S.IDSponsor)

                WHERE M.IDMagazin = id\_magazin;

                DBMS\_OUTPUT.PUT\_LINE('Suma salariilor: '||suma\_salarii||', Total suma investitie: '||total\_suma\_investitie);

                IF (suma\_salarii / 3) <= total\_suma\_investitie THEN-- Conditia de acoperire cheltuieli salarii

                    DBMS\_OUTPUT.PUT\_LINE('Suma salariilor angajatilor din magazinul curent este acoperita mai mult de o treime din suma de investitie');

                ELSE

                    DBMS\_OUTPUT.PUT\_LINE('Suma salariilor angajatilor din magazinul curent NU este acoperita mai mult de o treime din suma de investitie');

                END IF;

            END IF;

        END LOOP;

        CLOSE c\_angajati;

    END Afisare\_Angajati\_Cheltuieli\_Acoperite\_Magazin;

END Pachet\_Angajati;

/

BEGIN

    Pachet\_Angajati.Afisare\_Angajati\_Cheltuieli\_Acoperite\_Magazin;

    INSERT INTO Magazine (IDMagazin, IDAdresa, IDOrar, IDSponsor, NumeMagazin, RentabilitateOra)

    VALUES (14, 14, 5, null, 'Pufic 14', 66.11);-- Magazinul a fost ales sa nu aiba sponsor

    INSERT INTO Angajati (IDAngajat, IDMagazin, Nume, Prenume, DataAngajare, Salariu)

    VALUES (27, 14, 'Baietica', 'Luminita', TO\_DATE('2022-07-29', 'YYYY-MM-DD'), 3600.00);-- Un angajat de la magazinul fara sponsor

    IF Pachet\_Angajati.Verifica\_Sponsor\_Magazin(27) THEN-- TRUE

        DBMS\_OUTPUT.PUT\_LINE('Angajatul adaugat lucreaza la un magazin sponsorizat');

    ELSE-- FALSE

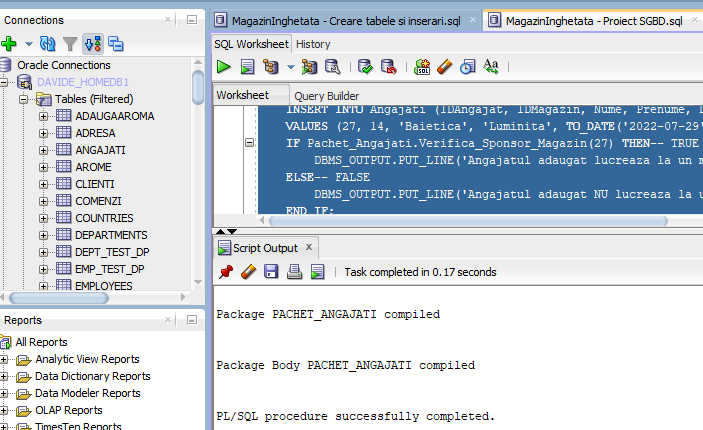
        DBMS\_OUTPUT.PUT\_LINE('Angajatul adaugat NU lucreaza la un magazin sponsorizat');

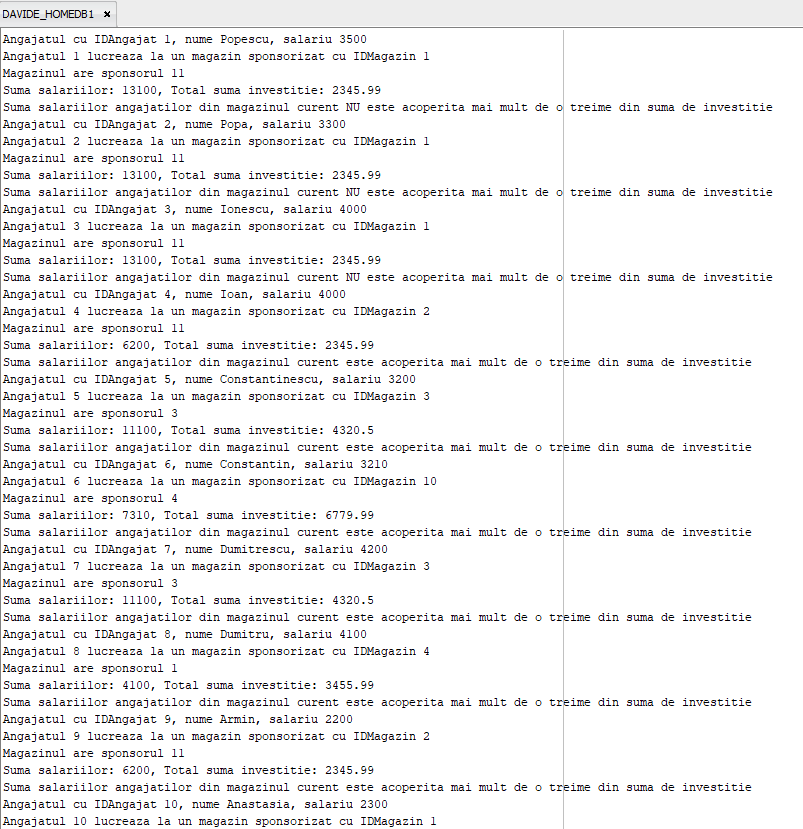
    END IF;

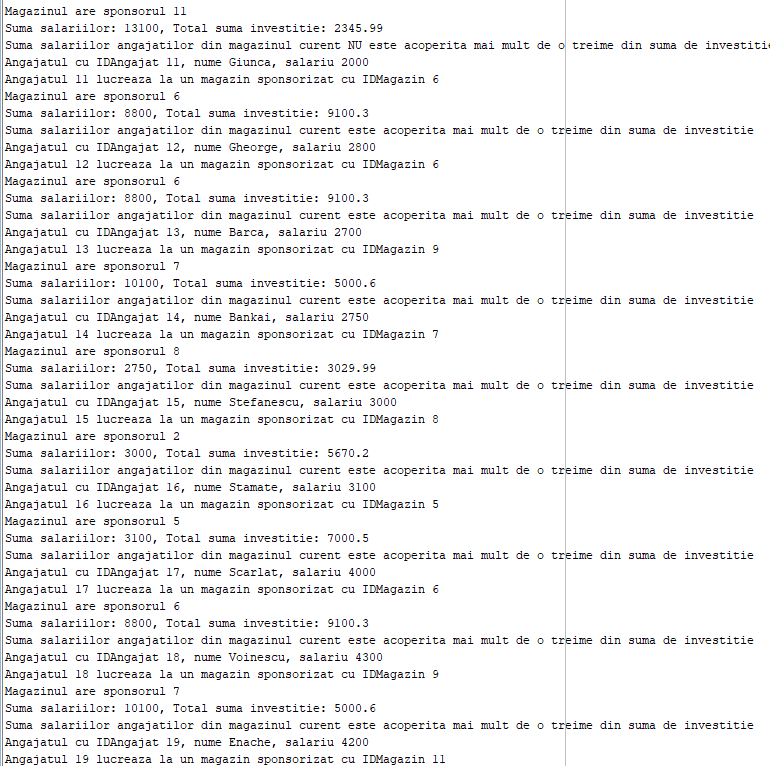
    ROLLBACK;

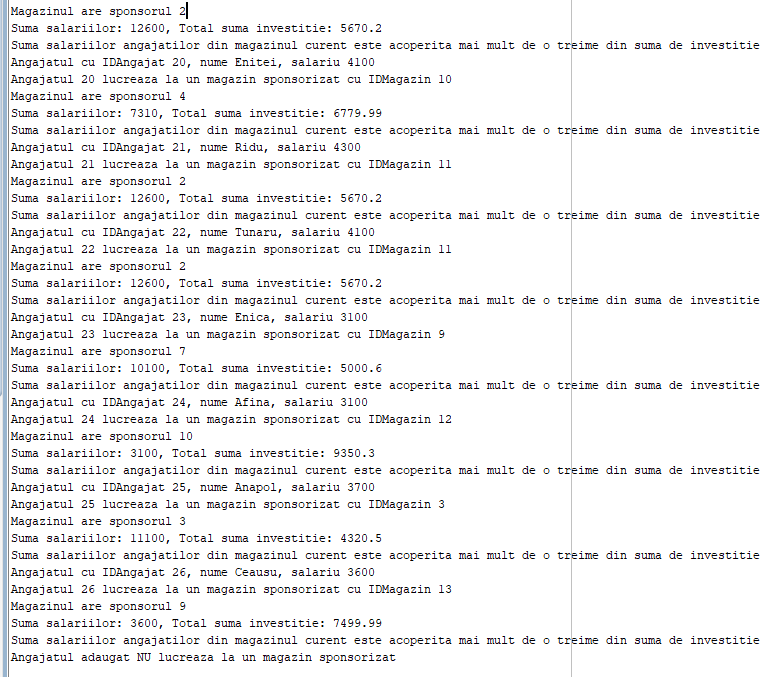
END;

/









# Diagrama Entitate Relație

ANGAJATI

1

SPONSOR

1

sponsorizeaza

M

M

1

detine

ORARMAGAZINE

1

MAGAZINE

1

are

1

ADRESA

M

M

depune

CLIENTI

1

FURNIZORI

aprovizioneaza

M

AROME

M

M

COMENZI

M

M

M

1

are

AdaugaAroma

RECENZII

lucreaza

Vinde

1

# Diagrama Conceptuală

CLIENTI

PK: IDClient

Prenume

Nume

Email

NumarTelefon



COMENZI

PK: IDComanda

DataComanda

TipPlata

VANZARI

PK, FK1: IDClient

PK, FK2: IDAngajat

PK, FK3: IDComanda

DataVanzare

AROME

PK: IDAroma

FK: IDFurnizor

Pret

NumeAroma

RECENZII

PK: IDRecenzie

FK: IDClient

Rating

Comentariu

DataRecenzie



ADAUGAAROMA

PK, FK1: IDComanda

PK, FK2: IDAroma

Cantitate



ANGAJATI

PK: IDAngajat

FK: IDMagazin

Nume

Prenume

DataAngajare

Salariu



ORARMAGAZINE

PK: IDOrar

OraDeschidere

OraInchidere

FURNIZORI

PK: IDFurnizor

NumeFurnizor

Email

NumarTelefon

ADRESA

PK: IDAdresa

Oras

Strada

SPONSOR

PK: IDSponsor

FK: IDAdresa

Nume

Email

NumarTelefon

SumaInvestitie

NumarEvenimente

MAGAZINE

PK: IDMagazin

FK1: IDAdresa

FK2: IDOrar

FK3: IDSponsor

NumeMagazin

RentabilitateOra

# Scheme relaționale

* MAGAZINE (IDMagazin #, NumeMagazin, RentabilitateOra, IDAdresa, IDOrar, IDSponsor)
* ORARMAGAZINE (IDOrar #, OraDeschidere, OraInchidere)
* SPONSOR (IDSponsor #, Nume, Email, NumarTelefon, SumaInvestitie, NumarEvenimente, IDAdresa)
* ADRESA (IDAdresa #, Oras, Strada)
* ANGAJATI (IDAngajat #, Nume, Prenume, DataAngajare, Salariu, IDMagazin)
* CLIENTI (IDClient #, Prenume, Nume, Email, NumarTelefon)
* RECENZII (IDRecenzie #, Rating, Comentariu, DataRecenzie, IDClient)
* COMENZI (IDComanda #, DataComanda, TipPlata)
* AROME (IDAroma #, Pret, NumeAroma, IDFurnizor)
* FURNIZORI (IDFurnizor #, NumeFurnizor, Email, NumarTelefon)
* ADAUGACOMANDA(IDComanda #, IDAroma #, Cantitate)
* VANZARI(IDClient #, IDAngajat #, IDComanda #, DataVanzare)

# Normalizări

Pentru reducerea redundanțelor datelor și eliminarea caracteristicilor precum anomalii la inserare, actualizare și ștergere folosim normalizări.

FN1:

Presupunem că entitatea Angajati nu ar fi avut o cheie primară unică, ceea ce înseamnă că este posibil să existe doi angajați cu aceleași atribute (Nume, Prenume, DataAngajare, Salariu). Din acest motiv, trebuie creată o nouă cheie primară pentru entitate.

FN2 :

Ca diagrama să fie în FN2, ea se află în FN1 și fiecare atribut care nu este cheie primară este dependent de întreaga cheie primară. Dacă entitatea Comanda ar avea în plus față de atributele definite, atribute de tipul Aroma1, Aroma2, ..., ar însemna că nu toate aceste atribute vor depinde de IDMagazin, deoarece e posibil să existe mai multe comenzi care să aibă aceeași combinație de arome de înghețată, fapt redundant. În plus, nu e recomandat să fie stabilit un număr fix de atribute de tipul acesta, deoarece unele înregistrări din Comanda vor avea mai puține arome față de altele. Pentru a rezolva această problemă, atributele acestea se elimină și apare o nouă entitate Arome și se stabilește o relație de many-to-many raportat la Comanda.

FN3:

Ca o relație să fie în FN3 trebuie să fie în FN2 și ca fiecare atribut ce diferă de o cheie să nu depindă de un alt atribut care la rândul lui nu este cheie. Toate atributele care nu sunt chei primare nu numai că sunt dependente de cheia primară, ci și independente unele de altele, eliminând astfel dependența tranzitivă.

Inițial, entitățile Magazine și Sponsor conțineau toate informațiile legate de adresă, ceea ce însemna că acestea conțineau atribute repetitive, în sensul că atributele Oras și Strada nu ar depinde direct nicin de IdMagazin și nici de IDSponsor. Din acest motiv, am creat o nouă entitate Adresa, care să rețină toate aceste informații, iar entitățile Magazine și Sponsor vor conține doar id-ul către entitatea Adresa, eliminând aparții care ar putea să se repete.