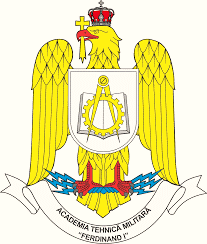
**PROIECT**

**BAZE DE DATE**

**Muzeul Sagrada**



Realizat de:

Std. Cap. Alexandru-Mihai Duminica

Grupa C112D

Contents

[1. Descrierea bazei de date 3](#_Toc134645703)

[1.1. Scopul proiectului 3](#_Toc134645704)

[1.2. Context 3](#_Toc134645705)

[1.3. Crearea bazei de date 4](#_Toc134645706)

[1.4. Crearea tabelelor 5](#_Toc134645707)

[1.5. Constrângeri 10](#_Toc134645708)

[1.6. Diagrama bazei de date 19](#_Toc134645709)

[2. Insert 19](#_Toc134645710)

[2.1. INSERT-Popularea bazei de date 19](#_Toc134645711)

[2.2. SELECT 40](#_Toc134645712)

[2.3. UPDATE 70](#_Toc134645713)

[2.4. DELETE 95](#_Toc134645714)

[2.5. Tranzacții 106](#_Toc134645715)

[2.6. CTE-uri 126](#_Toc134645716)

[3. Obiecte ale bazei de date 131](#_Toc134645717)

[3.1. VIEW-uri 131](#_Toc134645718)

[3.2. Proceduri stocate 136](#_Toc134645719)

[3.3. Trigger-e 145](#_Toc134645720)

# 1. Descrierea bazei de date

## 1.1. Scopul proiectului

Organizarea si functionarea adecvata a unui muzeu modern care s-a intalnit pentru prima data cu tehnologia necesara mentinerii acestuia intr-o stare de evidenta foarte buna legata de angajati, salariile acestora, impozitele, uneltele puse la dispozitie de muzeu, cat si furniorii si desfasurarea unui proces online de stabilire a rezervarilor prin evidenta expozitiilor accesibile si clientilor doritori.

## 1.2. Context

Tabelele reprezentand necesarul unei bune desfasurari a activitatii sunt :

* Tabelul Angajati – tine evidenta asupra angajatilor , evidentiind date cu caracter personal ale acestora, cat si departamentul sau data de angajare/expirare contract.
* Tabelul Clienti – tine evidenta asupra clientilor care efectueaza rezervari si contine date ale acestora, cat si tipul rezervarii.
* Tabelul Creatori – cuprinde date referitoare la persoanele care au contribuit cu creatiile sale la constituirea muzeului.
* Tabelul Exponate – cuprinde date referitoare la exponatele prezentate in cadrul Expozitiilor din muzeu, cat si expozitia din care acestea fac parte.
* Tabelul Expozitii – cuprinde detalii referitoare la expozitiile organizate in muzeu, cat si angajatul responsabil de acestea, fiecarui angajat revenindu-i o astfel de responsabilitate de participare activa alaturi de clienti.
* Tabelul Functii – cuprinde o detaliere a functiilor vitale pentru functionarea muzeului.
* Tabelul Furnizori – cuprinde detalii legate de furnizorii muzeului care pot fi atat de unelte, cat si de suveniruri.
* Tabelul Impozite – cuprinde detalii legate de taxele si impozitele impuse de stat muzeului.
* Tabelul Incasari – cuprinde detalii legate de sumele incasate de muzeu in timpul activitatii.
* Tabelul Rezervari – cuprinde detalii referitoare la rezervarile clientilor, precum id-ul clientului, numarul de bilete si id-ul expozitiei.
* Tabelul Salarii – cuprinde detalii referitoare la salariile angajatilor, precum data lichidare, salariu net, brut.
* Tabelul Suveniruri – cuprinde detalii referitoare la suvenirurile prezente in magazinul muzeului.
* Tabelul Unelte – cuprinde detalii referitoare la uneltele/materialele necesare bunei desfasurari a activitatii muzeului si care se afla in responsabilitatea angajatilor.
* Tabelul AngajatUnealta – tabel de legatura care leaga angajatii de uneltele asupra carora sunt responsabili.
* Tabelul CreatorExponat – tabel de legatura care leaga creatorii de exponatele create de acestia.
* Tabelul FunctieAngajat – tabelul de legatura care leaga Angajatii de functiile acestora in cadrul muzeului.
* Tabelul FurnizorSuvenir – tabelul de legatura care leaga Furnizorii de Suvenirurile produse de acestia.
* Tabelul FurnizorUnelte – tabelul de legatura care leaga Furnizorii de Uneltele produse de acestia.
* Tabelul IncasareRezervare – tabelul de legatura intre Incasari si Rezervari.
* Tabelul IncasareSuveniruri – tabelul de legatura intre Incasari si Suveniruri.
* Tabelul SalariuFunctie – tabelul de legatura intre Salarii si Functii.

## 1.3. Crearea bazei de date

**In cadrul creeri bazei de date, au fost introduce si instructiuni pentru crearea unor fisiere de date**

Pe lângă instrucțiunea de creare a bazei de date, se pot observa și instrucțiuni pentru crearea fișierelor de date(4) și a celor de jurnalizare a tranzacțiilor(2).

CREATE DATABASE MIHotels

ON PRIMARY

(

NAME = MuzeuSagradaData1,

FILENAME = 'D:\DB\MuzeuSagrada.mdf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

),

(

NAME = MuzeuSagradaData2,

FILENAME = 'D:\DB\MuzeuSagrada1.mdf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

),

(

NAME = MuzeuSagradaData3,

FILENAME = 'D:\DB\MuzeuSagrada.ndf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

),

(

NAME = MuzeuSagradaData4,

FILENAME = 'D:\DB\MuzeuSagrada1.ndf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

)

LOG ON

(

NAME = LoggerMuzeu1,

FILENAME = 'D:\DB\LoggerMuzeu1.ldf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

),

(

NAME = LoggerMuzeu2,

FILENAME = 'D:\DB\LoggerMuzeu2.ldf',

size = 10MB,

maxsize = UNLIMITED,

filegrowth = 1GB

);

## 1.4. Crearea tabelelor

if OBJECT\_ID('Angajati','U') is not null

drop table Angajati

create table Angajati(

IDAngajat int identity(1,1) primary key,

Nume nvarchar(20),

Prenume nvarchar(20),

Localitate nvarchar(20),

Departament nvarchar(20),

NumarTelefon char(10),

DataAngajare Date,

DataExpirareContract Date,

AniExperientaDomeniu int

)

if OBJECT\_ID('Functii','U') is not null

drop table Functii

create table Functii(

IDFunctie int identity(1,1) primary key,

NumeFunctie nvarchar(20),

IntervalOrar nvarchar(20),

Responsabilitati nvarchar(100),

Executie char(2),

DataObtinereFunctie DATE

)

if OBJECT\_ID('Salarii','U') is not null

drop table Salarii

create table Salarii(

IDSalariu int identity(1,1) primary key,

SalariuBrut int,

SalariuNet int,

DataAvans Date,

DataLichidare Date,

OreSuplimentare int,

ValoareAvans int

)

if OBJECT\_ID('SalariuFunctie','U') is not null

drop table SalariuFunctie

create table SalariuFunctie(

IDSalariuFunctie int identity(1,1) primary key,

IDSalariu int,

IDFunctie int,

foreign key(IDSalariu) references Salarii(IDSalariu),

foreign key(IDFunctie) references Functii(IDFunctie)

)

if OBJECT\_ID('FunctieAngajat','U') is not null

drop table FunctieAngajat

create table FunctieAngajat(

IDFunctieAngajat int identity(1,1) primary key,

IDFunctie int,

IDAngajat int,

DataObtinereFunctie DATE,

foreign key(IDFunctie) references Functii(IDFunctie),

foreign key(IDAngajat) references Angajati(IDAngajat)

)

if OBJECT\_ID('Impozite','U') is not null

drop table Impozite

create table Impozite(

IDImpozit int identity(1,1) primary key,

IDSalariu int,

ProcentAsigurareMedicala int,

ProcentCasaPensii int,

ProcentTaxaSalariat int,

foreign key(IDSalariu) references Salarii(IDSalariu)

)

if OBJECT\_ID('Furnizori','U') is not null

drop table Furnizori

create table Furnizori(

IDFurnizor int identity(1,1) primary key,

NumeFurnizor nvarchar(50),

CodSRL varchar(7),

Adresa nvarchar(50),

UltimaLivrare Date,

DurataContractFurnizare int,

AnActivitatePiata Date,

)

if OBJECT\_ID('Suveniruri','U') is not null

drop table Suveniruri

create table Suveniruri(

IDSuvenir int identity(1,1) primary key,

Nume nvarchar(20),

Material nvarchar(15),

Tip nvarchar(15),

TaraProvenienta nvarchar(20),

Pret int,

BucatiStoc int,

)

if OBJECT\_ID('FurnizorSuvenir','U') is not null

drop table FurnizorSuvenir

create table FurnizorSuvenir(

IDFurnizorSuvenir int identity(1,1) primary key,

IDSuvenir int,

IDFurnizor int,

foreign key(IDSuvenir) references Suveniruri(IDSuvenir) on cascade delete,

foreign key(IDFurnizor) references Furnizori(IDFurnizor) on cascade delete

)

if OBJECT\_ID('Unelte','U') is not null

drop table Unelte

create table Unelte(

IDUnealta int identity(1,1) primary key,

Nume nvarchar(20),

Marca nvarchar(20),

TipUnealta nvarchar(20),

DataAchizitie Date,

DataExpirareGarantie Date,

)

if object\_id('FurnizorUnealta','U') is not null

drop table FurnizorUnealta

create table FurnizorUnealta(

IDFurnizorUnealta int identity(1,1) primary key,

IDUnealta int,

IDFurnizor int,

foreign key(IDUnealta) references Unelte(IDUnealta) on cascade delete,

foreign key(IDFurnizor) references Furnizori(IDFurnizor) on cascade delete

)

if object\_id('AngajatUnealta','U') is not null

drop table AngajatUnealta

create table AngajatUnealta(

IDAngajatUnealta int identity(1,1) primary key,

IDAngajat int,

IDUnealta int,

foreign key(IDAngajat) references Angajati(IDAngajat) on cascade delete,

foreign key(IDUnealta) references Unelte(IDUnealta) on cascade delete

)

if object\_id('Incasari','U') is not null

drop table Incasari

create table Incasari(

IDIncasare int identity(1,1) primary key,

SumaIncasata int,

NumarClienti int

)

if object\_id('IncasareSuvenir','U') is not null

drop table IncasareSuvenir

create table IncasareSuvenir(

IDIncasareSuvenir int identity(1,1) primary key,

IDIncasare int,

IDSuvenir int,

DataIncasare int,

foreign key(IDIncasare) references Incasari(IdIncasare) on cascade delete,

foreign key(IDSuvenir) references Suveniruri(IDSuvenir) on cascade delete

)

if object\_id('Clienti','U') is not null

drop table Clienti

create table Clienti(

IDClient int identity(1,1) primary key,

Nume nvarchar(20),

Prenume nvarchar(20),

ClientFidel char(2),

TipClient nvarchar(10),

Telefon char(10),

Mail nvarchar(30),

DataUltimaVizita Date

)

if object\_id('Expozitii','U') is not null

drop table Expozitii

create table Expozitii(

IDExpozitie int identity(1,1) primary key,

IDAngajat int,

Locatie nvarchar(20),

Descriere nvarchar(100),

CapacitatePersoane int,

foreign key(IDAngajat) references Angajati(IDAngajat)

)

if object\_id('Rezervari','U') is not null

drop table Rezervari

create table Rezervari(

IDRezervare int identity(1000,1) primary key,

IDExpozitie int,

IDClient int,

NumarBilete int,

DataRezervare Date,

OraRezervare int,

foreign key(IDClient) references Clienti(IDClient),

foreign key(IDExpozitie) references Expozitii(IDExpozitie)

)

if object\_id('Exponate','U') is not null

drop table Exponate

create table Exponate(

IDExponat int identity(1,1) primary key,

IDExpozitie int,

DataAparitie Date,

Nume nvarchar(20),

Tip nvarchar(20),

TaraAparitie nvarchar(10),

Descriere nvarchar(50),

foreign key(IDExpozitie) references Expozitii(IDExpozitie)

)

if object\_id('Creatori','U') is not null

drop table Creatori

create table Creatori(

IDCreator int identity(1,1) primary key,

Nume nvarchar(20),

Prenume nvarchar(20),

DataNastere Date,

DataDeces Date,

Adresa nvarchar(30),

Studii nvarchar(20),

TipuriPreferate nvarchar(30)

)

if object\_id('CreatorExponat','U') is not null

drop table CreatorExponat

create table CreatorExponat(

IDCreatorExponat int identity(1,1) primary key,

IDCreator int,

IDExponat int,

foreign key(IDCreator) references Creatori(IDCreator),

foreign key(IDExponat) references Exponate(IDExponat)

)

if object\_id('IncasareRezervare','U') is not null

drop table IncasareRezervare

create table IncasareRezervare(

IDIncasareRezervare int identity(1,1) primary key,

IDIncasare int,

IDRezervare int,

DataIncasare Date,

foreign key(IDIncasare) references Incasari(IDIncasare),

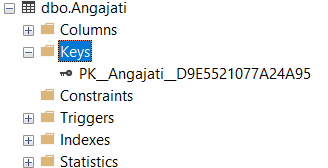
foreign key(IDRezervare) references Rezervari(IDRezervare)

)

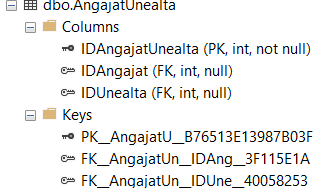
## 1.5. Constrângeri

Constrangerile de cheie primara/straina asigura legaturile intre anumite tabele ale bazei de date si sunt prezentate in cele ce urmeaza:

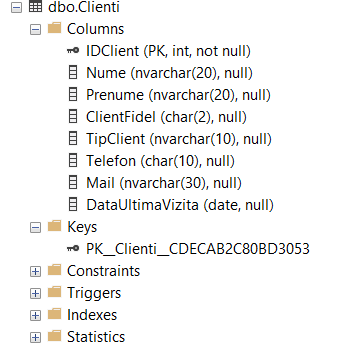
1.Tabelul Angajati



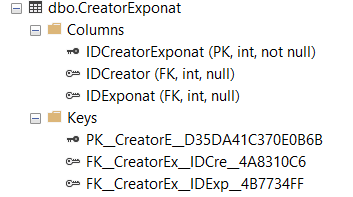
2. Tabelul AngajatUnealta



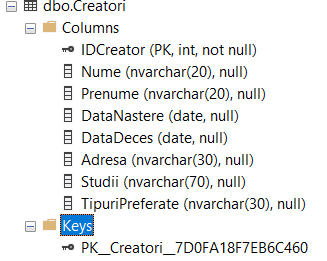
3. Tabelul Clienti



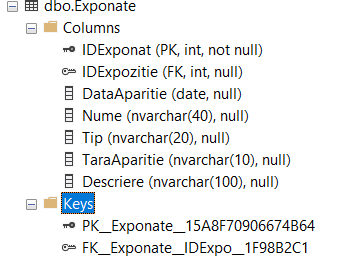
4. Tabelul CreatorExponat



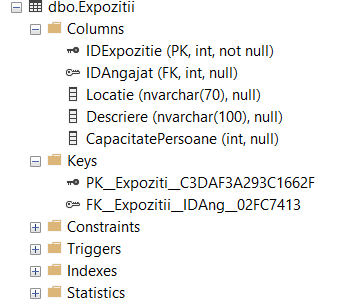
5. Tabelul Creatori



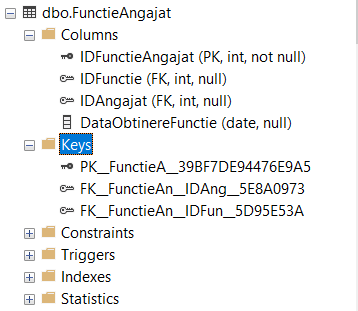
6. Tabelul Exponate



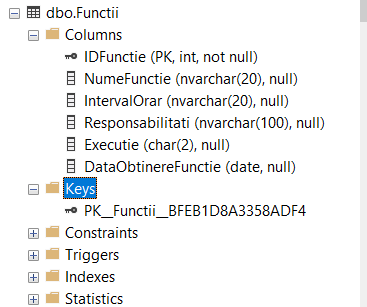
7. Tabelul Expozitii



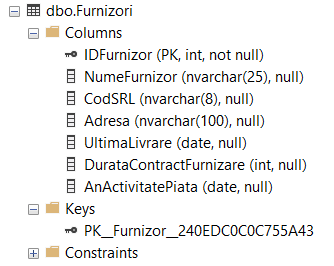
8. Tabelul FunctieAngajat



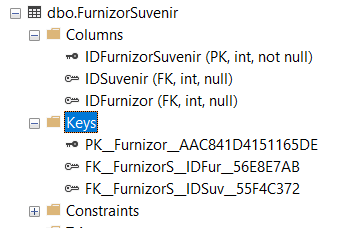
9. Tabelul Functii



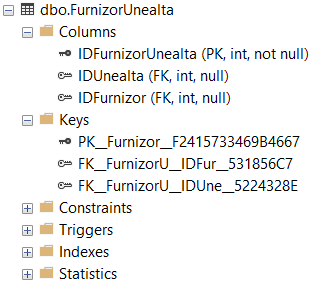
10. Tabelul Furnizori



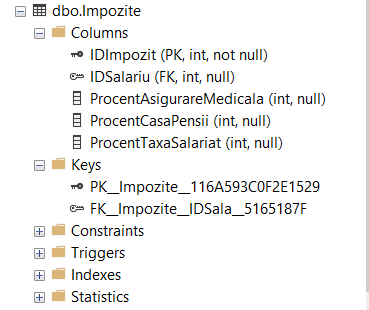
11. Tabelul FurnizorSuvenir



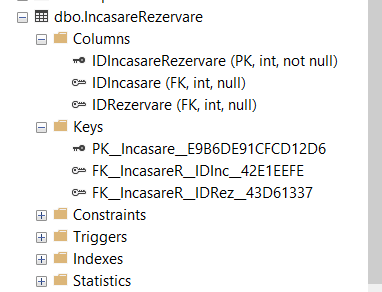
12. Tabelul FurnizorUnealta



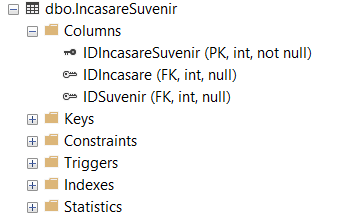
13. Tabelul Impozite



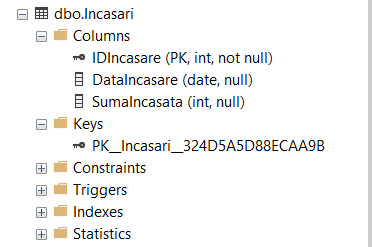
14. Tabelul IncasareRezervare



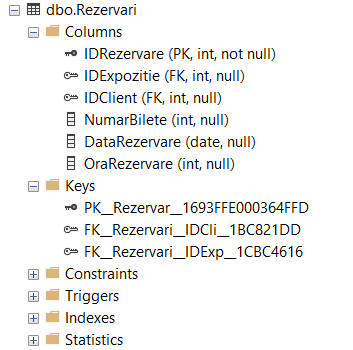
15. Tabelul IncasareSuvenir



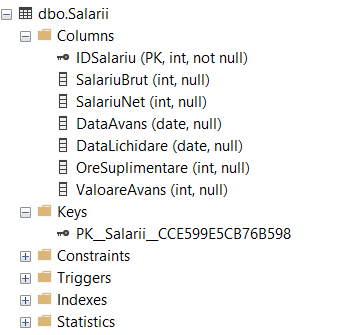
16. Tabelul Incasari



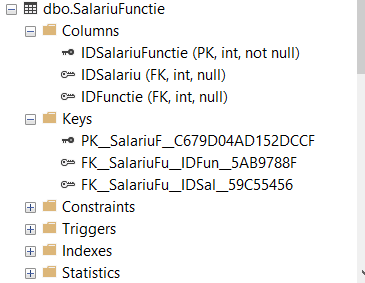
17. Tabelul Rezervari



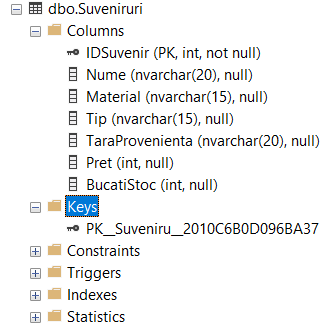
18. Tabelul Salarii



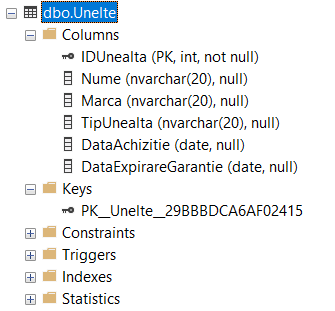
19. Tabelul SalariuFunctie



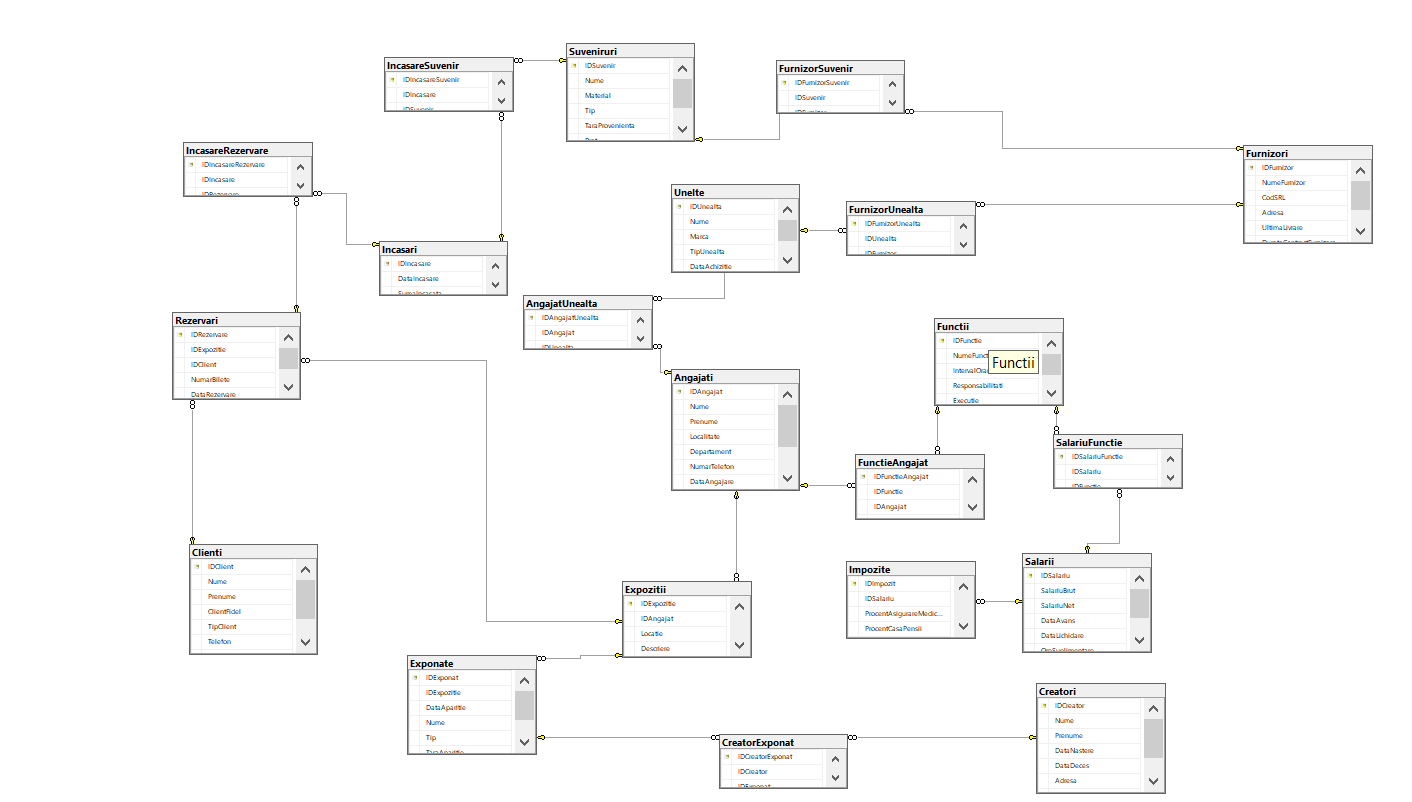
20. Tabelul Suveniruri



21. Tabelul Unelte



## 1.6. Diagrama bazei de date

****

# 2. Insert

## 2.1. INSERT-Popularea bazei de date

**1. Tabelul Angajati**

INSERT INTO Angajati (Nume, Prenume, Localitate, Departament, NumarTelefon, DataAngajare, DataExpirareContract, AniExperientaDomeniu)

VALUES

('Popescu', 'Ion', 'Bucuresti', 'IT', '0756215487', '2022-01-15', '2022-07-15', 2),

('Ionescu', 'Ana', 'Iasi', 'HR', '0732458963', '2021-09-01', '2023-09-01', 5),

('Vasilescu', 'Andrei', 'Brasov', 'Vanzari', '0756654321', '2022-03-02', '2023-03-02', 3),

('Georgescu', 'Mihai', 'Constanta', 'Ingrijire', '0721122334', '2022-02-28', '2022-08-28', 1),

('Dragomirescu', 'Elena', 'Ploiesti', 'Ingrijire', '0766112233', '2022-01-01', '2022-07-01', 4),

('Dumitru', 'Alexandru', 'Bucuresti', 'Primire oaspeti', '0734112233', '2022-04-15', '2022-10-15', 2),

('Popa', 'Maria', 'Cluj-Napoca', 'Vanzari', '0722334455', '2022-01-05', '2023-01-05', 6),

('Marin', 'Andreea', 'Timisoara', 'Primire oaspeti', '0754112233', '2022-01-20', '2022-07-20', 1),

('Gheorghe', 'Diana', 'Oradea', 'Management', '0721002233', '2022-02-01', '2023-02-01', 3),

('Voicu', 'Razvan', 'Bacau', 'Vanzari', '0766888822', '2022-03-15', '2022-09-15', 2),

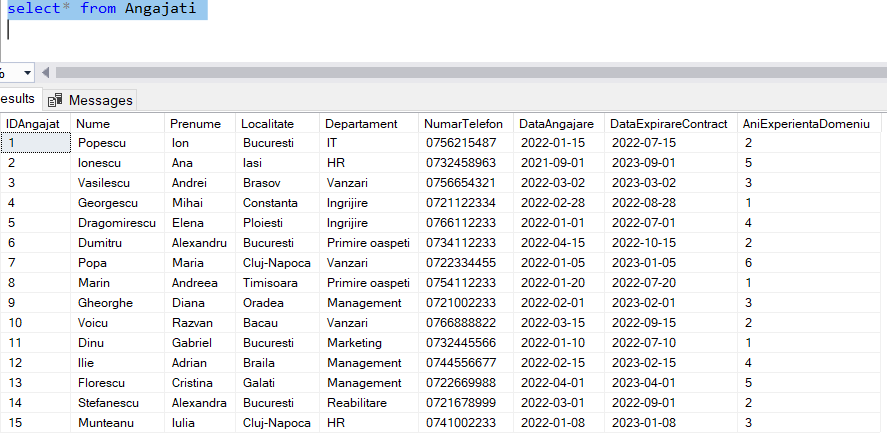
('Dinu', 'Gabriel', 'Bucuresti', 'Marketing', '0732445566', '2022-01-10', '2022-07-10', 1),

('Ilie', 'Adrian', 'Braila', 'Management', '0744556677', '2022-02-15', '2023-02-15', 4),

('Florescu', 'Cristina', 'Galati', 'Management', '0722669988', '2022-04-01', '2023-04-01', 5),

('Stefanescu', 'Alexandra', 'Bucuresti', 'Reabilitare', '0721678999', '2022-03-01', '2022-09-01', 2),

('Munteanu', 'Iulia', 'Cluj-Napoca', 'HR', '0741002233', '2022-01-08', '2023-01-08', 3);

****

**2. Tabelul Functii**

INSERT INTO Functii (NumeFunctie, IntervalOrar, Responsabilitati, Executie, DataObtinereFunctie)

VALUES

('Ghid turistic', '9:00 - 17:00', 'Conduce vizitatorii prin muzeu si le ofera informatii despre exponate', 'Da', '2022-01-01'),

('Curator', '9:00 - 18:00', 'Coordoneaza activitatile de cercetare si expozitionare ale muzeului', 'Da', '2021-05-01'),

('Arhivar', '9:00 - 16:30', 'Se ocupa de organizarea si mentinerea in bune conditii a arhivei muzeului', 'Nu', '2021-09-15'),

('Conservator', '10:00 - 18:00', 'Restaureaza obiectele de arta si de patrimoniu din colectia muzeului', 'Da', '2020-06-01'),

('Asistent curator', '9:00 - 17:00', 'Ofera suport curatorului in activitatile de cercetare si expozitionare ale muzeului', 'Da', '2022-02-01'),

('Tehnician', '9:00 - 17:00', 'Se ocupa de mentenanta si reparatia echipamentelor si sistemelor muzeului', 'Nu', '2021-04-01'),

('Administrator', '9:00 - 17:00', 'Se ocupa de administrarea si gestionarea resurselor muzeului', 'Nu', '2020-01-01'),

('Manager', '9:00 - 18:00', 'Coordoneaza activitatile operationale si strategice ale muzeului', 'Nu', '2019-05-01'),

('Asistent cercetare', '9:00 - 17:00', 'Ofera suport cercetatorilor in activitatile de cercetare ale muzeului', 'Da', '2022-03-01'),

('Designer expozitie', '9:00 - 18:00', 'Se ocupa de designul si organizarea expozitiilor din muzeu', 'Da', '2021-10-01'),

('Vanzator bilete', '9:00 - 17:00', 'Ofera bilete si informatii turistilor care viziteaza muzeul', 'Nu', '2022-04-01'),

('Receptioner', '9:00 - 17:00', 'Se ocupa de primirea si redirectionarea vizitatorilor catre sectiunile muzeului', 'Nu', '2020-09-01'),

('Sofer', '9:00 - 17:00', 'Conduce vehiculele muzeului si transporta obiectele de patrimoniu', 'Nu', '2021-03-01');

****

**3. Tabel Salarii**

INSERT INTO Salarii (SalariuBrut, SalariuNet, DataAvans, DataLichidare, OreSuplimentare, ValoareAvans)

VALUES

(4000, 3200, '2023-04-01', '2023-04-30', 10, 500),

(3000, 2400, '2023-04-01', '2023-04-30', 8, 200),

(3500, 2800, '2023-04-01', '2023-04-30', 12, 300),

(4500, 3600, '2023-04-01', '2023-04-30', 6, 400),

(5000, 4000, '2023-04-01', '2023-04-30', 14, 600),

(2500, 2000, '2023-04-01', '2023-04-30', 5, 100),

(4200, 3360, '2023-04-01', '2023-04-30', 15, 700),

(3200, 2560, '2023-04-01', '2023-04-30', 11, 400),

(3700, 2960, '2023-04-01', '2023-04-30', 9, 250),

(2800, 2240, '2023-04-01', '2023-04-30', 7, 150),

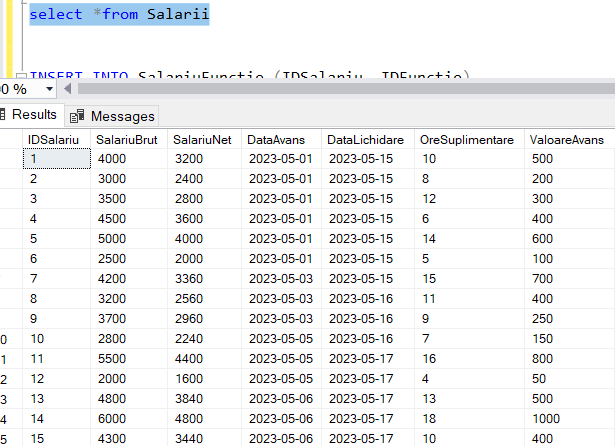
(5500, 4400, '2023-04-01', '2023-04-30', 16, 800),

(2000, 1600, '2023-04-01', '2023-04-30', 4, 50),

(4800, 3840, '2023-04-01', '2023-04-30', 13, 500),

(6000, 4800, '2023-04-01', '2023-04-30', 18, 1000),

(4300, 3440, '2023-04-01', '2023-04-30', 10, 400);

****

**4. Tabelul SalariuFunctie**

INSERT INTO SalariuFunctie (IDSalariu, IDFunctie)

VALUES

(1, 1),

(2, 3),

(3, 2),

(4, 1),

(5, 4),

(6, 1),

(7, 2),

(8, 4),

(9, 3),

(10, 1),

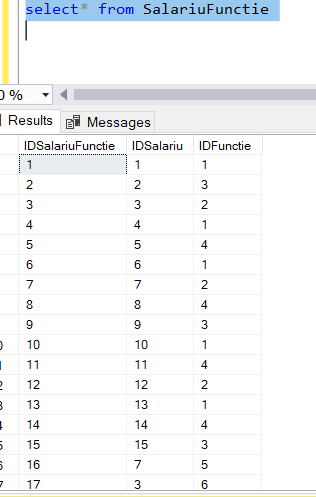
(11, 4),

(12, 2),

(13, 1),

(14, 4),

(15, 3);

****

**5. Tabelul Impozite**

INSERT INTO Impozite (IDSalariu, ProcentAsigurareMedicala, ProcentCasaPensii, ProcentTaxaSalariat)

VALUES

(1,5 , 7, 3),

(2, 4, 3, 10),

(3, 7, 9, 3),

(4, 10, 9, 3),

(5, 10, 15, 5),

(6, 10, 15, 5),

(7, 10, 20, 14),

(8, 3, 10, 2),

(9, 7, 5, 3),

(10, 5, 5, 3),

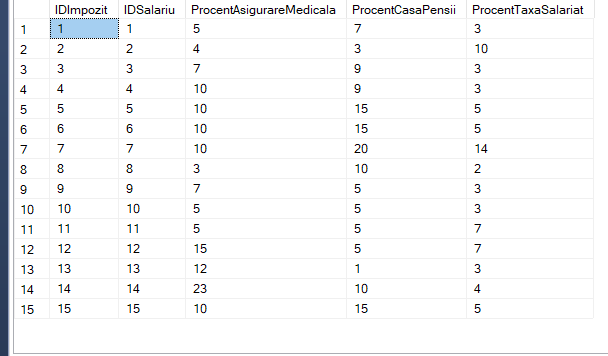
(11, 5, 5, 7),

(12, 15, 5, 7),

(13, 12, 1, 3),

(14, 23, 10, 4),

(15, 10, 15, 5);

****

**6. Tabelul Furnizori**

Insert into Furnizori(NumeFurnizor, CodSRL, Adresa, UltimaLivrare, DurataContractFurnizare, AnActivitatePiata) values

('SC Prodcom SRL', 'RO5412', 'Strada Mare, nr. 25, Bucuresti', '2021-10-20', 12, '2010-01-01'),

('Firma Global SRL', 'RO54321', 'Strada Pietei, nr. 10, Cluj-Napoca', '2022-03-15', 6, '2015-06-01'),

('Sofistic Art SRL', 'RO67890', 'Strada Florilor, nr. 7, Timisoara', '2022-01-05', 24, '2005-12-15'),

('Best Trade SRL', 'RO24680', 'Bulevardul Libertatii, nr. 1, Iasi', '2021-12-01', 18, '2008-09-30'),

('Mega Prod SRL', 'RO13579', 'Strada Victoriei, nr. 20, Cluj-Napoca', '2021-11-10', 12, '2012-05-10'),

('Arti Gold SRL', 'RO98765', 'Bulevardul Unirii, nr. 12, Bucuresti', '2022-02-20', 6, '2019-04-01'),

('New Art SRL', 'RO01234', 'Strada Muncii, nr. 5, Timisoara', '2022-01-30', 12, '2003-08-01'),

('Decor Art SRL', 'RO56789', 'Strada Principala, nr. 30, Brasov', '2021-12-15', 24, '2006-02-15'),

('Classic Style SRL', 'RO24601', 'Bulevardul Stefan cel Mare, nr. 5, Iasi', '2022-03-25', 12, '2011-11-01'),

('Gift Shop SRL', 'RO13579', 'Strada Unirii, nr. 15, Cluj-Napoca', '2022-02-01', 18, '2007-05-01'),

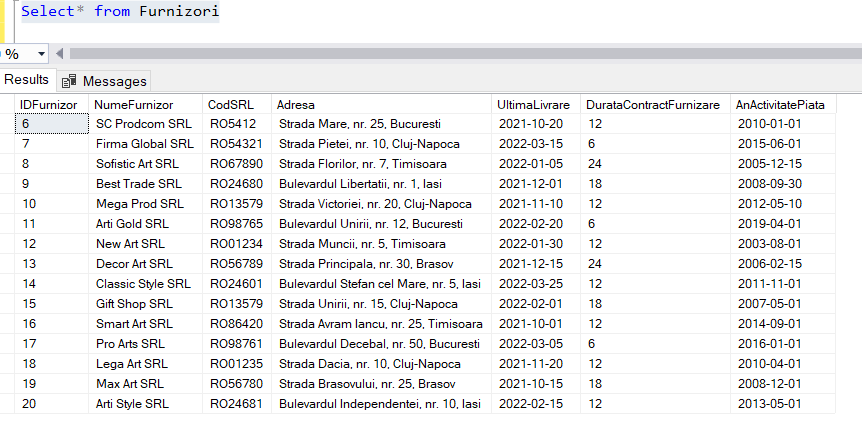
('Smart Art SRL', 'RO86420', 'Strada Avram Iancu, nr. 25, Timisoara', '2021-10-01', 12, '2014-09-01'),

('Pro Arts SRL', 'RO98761', 'Bulevardul Decebal, nr. 50, Bucuresti', '2022-03-05', 6, '2016-01-01'),

('Lega Art SRL', 'RO01235', 'Strada Dacia, nr. 10, Cluj-Napoca', '2021-11-20', 12, '2010-04-01'),

('Max Art SRL', 'RO56780', 'Strada Brasovului, nr. 25, Brasov', '2021-10-15', 18, '2008-12-01'),

('Arti Style SRL', 'RO24681', 'Bulevardul Independentei, nr. 10, Iasi', '2022-02-15', 12, '2013-05-01')

****

**7. Tabelul Suveniruri**

INSERT INTO Suveniruri (Nume, Material, Tip, TaraProvenienta, Pret, BucatiStoc) VALUES

('Magnet Frigider', 'Metal', 'Magnet', 'Romania', 5, 10),

('Tricou muzeu', 'Bumbac', 'Tricou', 'Romania', 20, 5),

('Breloc cheie masina', 'Metal', 'Breloc', 'Franta', 8, 15),

('Magnet Muzeu', 'Metal', 'Magnet', 'Romania', 6, 20),

('Pahar Negru-mat', 'Sticla', 'Pahar', 'Cehia', 12, 10),

('Breloc', 'Metal', 'Breloc', 'Italia', 7, 25),

('Tricou NY', 'Bumbac', 'Tricou', 'Spania', 18, 8),

('Pahar Muzeu', 'Sticla', 'Pahar', 'Germania', 11, 12),

('Termometru', 'Metal', 'Activ', 'Olanda', 5, 15),

('Breloc Oras', 'Metal', 'Breloc', 'Marea Britanie', 9, 20),

('Pahar Viena', 'Sticla', 'Pahar', 'Austria', 13, 7),

('Magnet New York', 'Metal', 'Magnet', 'SUA', 7, 18),

('Tricou Los Angeles', 'Bumbac', 'Tricou', 'SUA', 22, 6),

('Breloc Sidney', 'Metal', 'Breloc', 'Australia', 10, 10),

('Sculptura Muzeu', 'Lemn', 'Pahar', 'Brazilia', 15, 9),

('Sculptura Muzeu', 'Lemn', 'Aranjament', 'Brazilia', 15, 9),

('Magnet copil', 'Metal', 'Magnet', 'Romania', 5, 10),

('Hanorac muzeu', 'Bumbac', 'Vestimentar', 'Romania', 20, 5),

('Breloc model nou', 'Metal', 'Breloc', 'Franta', 8, 15),

('Magnet LV', 'Metal', 'Magnet', 'Romania', 6, 20),

('Pahar colorat', 'Sticla', 'Pahar', 'Cehia', 12, 10),

('Vedere', 'Metal', 'Poza', 'Italia', 7, 25),

('Tricou verde NY', 'Bumbac', 'Tricou', 'Spania', 18, 8),

('Pahar Muzeu Versace', 'Sticla', 'Pahar', 'Germania', 11, 12),

('Magnet chei', 'Metal', 'Magnet', 'Olanda', 5, 15),

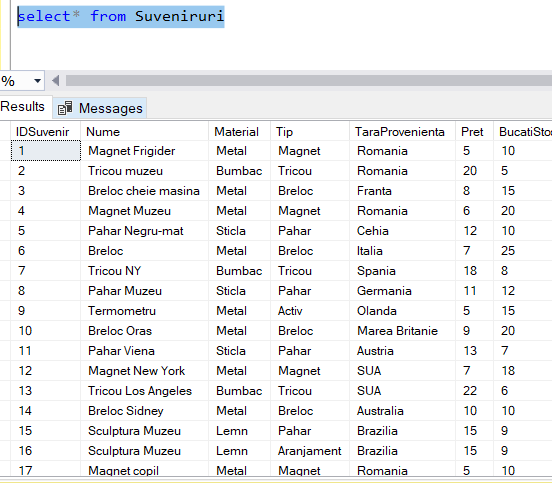
('Breloc Oras', 'Metal', 'Breloc', 'Marea Britanie', 9, 20),

('Cana colorata', 'Sticla', 'Cana', 'Austria', 13, 7),

('Magnet New York', 'Metal', 'Magnet', 'SUA', 7, 18),

('Tricou caricatura', 'Bumbac', 'Tricou', 'SUA', 22, 6),

('Breloc Sidney', 'Metal', 'Breloc', 'Australia', 10, 10);

****

**8. Tabelul FurnizorSuvenir**

INSERT INTO FurnizorSuvenir(IDSuvenir, IDFurnizor)

VALUES (1, 9),

(2, 10),

(3, 11),

(4, 11),

(5, 7),

(6, 8),

(7, 10),

(8, 10),

(9, 11),

(10, 12),

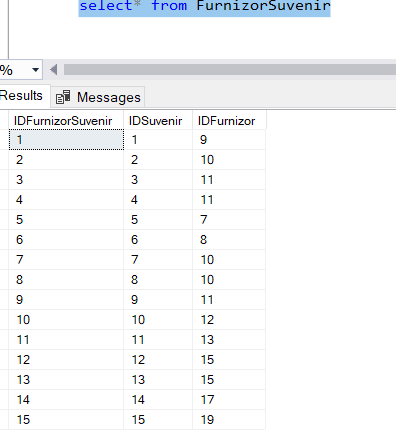
(11, 13),

(12, 15),

(13, 15),

(14, 17),

(15, 19);

****

**9. Tabelul Unelte**

INSERT INTO Unelte(Nume, Marca, TipUnealta, DataAchizitie, DataExpirareGarantie)

VALUES ('Unitate PC', 'Samsung', 'Electric', '2020-01-01', '2023-01-01'),

('Monitor', 'Dell', 'Electric', '2021-02-05', '2025-02-05'),

('Kit programare', 'Lenovo', 'Electric', '2019-06-15', '2022-06-15'),

('Monitor', 'Dell', 'Electric', '2022-03-20', '2025-03-20'),

('Unitate PC', 'Dell', 'Electric', '2020-12-01', '2024-12-01'),

('Intrerupator', 'Schneider', 'Electric', '2021-09-10', '2026-09-10'),

('Casa de marcat', 'Phillips', 'Electric', '2019-04-12', '2022-04-12'),

('Polizor unghiular', 'Makita', 'Electric', '2020-07-20', '2023-07-20'),

('Cleste', 'Knipex', 'Manual', '2021-01-10', '2024-01-10'),

('Fierastrau circular', 'Bosch', 'Electric', '2022-02-14', '2027-02-14'),

('Lampa de lucru', 'Einhell', 'Electrica', '2020-05-01', '2023-05-01'),

('Aparat evidenta', 'LoveON', 'Electric', '2021-08-10', '2026-08-10'),

('Aparat monitorizare', 'LCD', 'Bruiaj', '2019-07-08', '2022-07-08'),

('Costum special', 'Bigotti', 'Vestimentar', '2022-04-15', '2025-04-15'),

('Prelungitor', 'Hammas', 'Electric', '2020-11-10', '2023-11-10'),

('Calculator buzunar','Dell','Electric','2020/01/01','2023/01/01'),

('Masina de gaurit', 'DeWalt' ,'Electric' ,'2021-02-05' ,'2025-02-05'),

('Imprimanta Dell', 'Recuzita' ,'2019-06-15','2022-06-15'),

('Foarfeca Fiskars', 'Manuala','2022-03-20' ,'2025-03-20'),

('PC Monitorizare', 'Dell' ,'Electric','2020-12-01', '2024-12-01'),

('Tabla portabila', 'Ineos' ,'Recuzita' ,'2021-09-10' ,'2026-09-10'),

('Server Synology', 'Electric' ,'2019-04-12', '2022-04-12'),

('Server SynologyS', ',Electric' ,'2020-07-20' ,'2023-07-20'),

('Cleste Knipex', 'Manual', '2021-01-10' ,'2024-01-10'),

('Imprimanta Hama', 'Recuzita' ,'2022-02-14', '2027-02-14'),

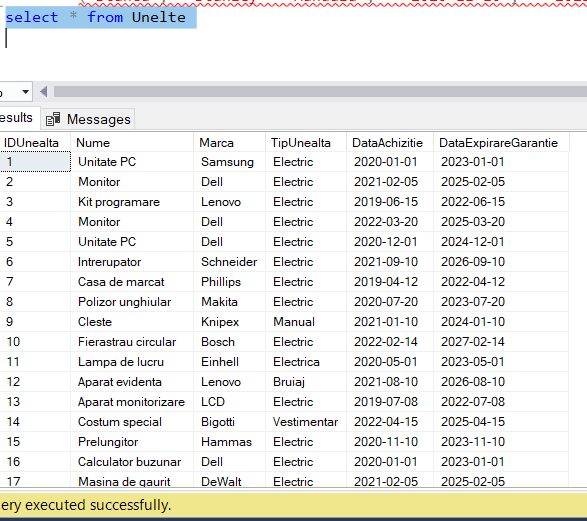
('Lampa de lucru', 'Einhell' ,'Electrica' ,'2020-05-01' ,'2023-05-01'),

('Tabla Smart', 'Hama' ,'Electric' ,'2021-08-10' ,'2026-08-10'),

('Markere', 'Bic', 'Recuzita', '2019-07-08', '2022-07-08'),

('Imprimanta', 'Canon Recuzita', '2022-04-15' ,'2025-04-15'),

('Stanca', 'Stanley Manuala', '2020-11-10', '2023-11-10');

****

**10. Tabelul FurnizorUnealta**

INSERT INTO FurnizorUnealta(IDUnealta, IDFurnizor)

VALUES (1, 6),

(2, 8),

(3, 9),

(4, 10),

(5, 6),

(6, 7),

(7, 12),

(8, 13),

(9, 14),

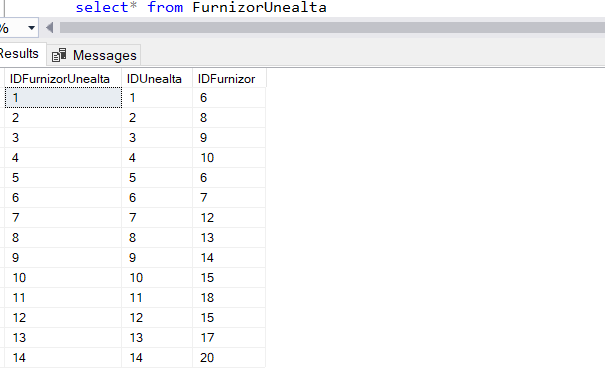
(10, 15),

(11, 18),

(12, 15),

(13, 17),

(14, 20);

****

**11. Tabelul AngajatUnealta**

insert into AngajatUnealta(IDAngajat,IDUnealta) values

(1, 1),

(1, 2),

(1, 3),

(2, 4),

(2, 5),

(2, 6),

(3, 4),

(4, 8),

(4, 9),

(4, 10),

(5, 4),

(5, 10),

(5, 11),

(6, 12),

(6, 13),

(6,14),

(7,16),

(7,18),

(8,12),

(8,14),

(9,18),

(9,21),

(1,22),

(9,23),

(1,23),

(9,25),

(12,25),

(13,25),

(15,25),

(15,27),

(12,27),

(12,28),

(13,28),

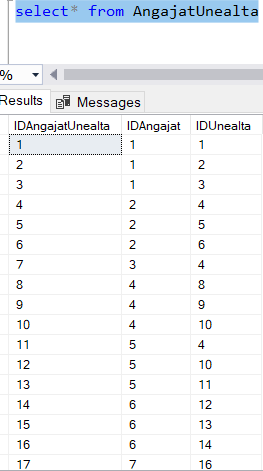
(10,28),

(10,29),

(5,24),

(5,19),

(6,19);

****

**12. Tabelul Incasari**

insert into Incasari (SumaIncasata,NumarClienti) values

(100,20),

(200,30),

(300,10),

(400,5),

(500,40),

(600,12),

(700,15),

(800,30),

( 900,25),

(1000,47),

( 1100,42),

( 1200,39),

(1300,25),

( 1400,36),

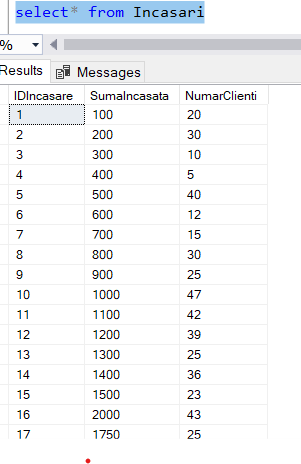
( 1500,23),

(2000,43),

(1750,25),

(2000,100),

(2100,153);

****

**13. Tabel IncasareSuvenir**

insert into IncasareSuvenir(IDIncasare,IDSuvenir,DataIncasare) values

(1, 1,'2023/05/01'),

(1, 2,'2022/05/02'),

(2, 3,'2023/04/05'),

(2, 4,'2023/03/27'),

(3, 5,'2023/05/23'),

(3, 6,'2023/05/20'),

(4, 7,'2022/04/25'),

(4, 8,'2022/08/05'),

(5, 9,'2023/07/05'),

(5, 10,'2023/05/04'),

(6, 11,'2022/04/03'),

(6, 12,'2021/07/03'),

(7, 13,'2022/05/29'),

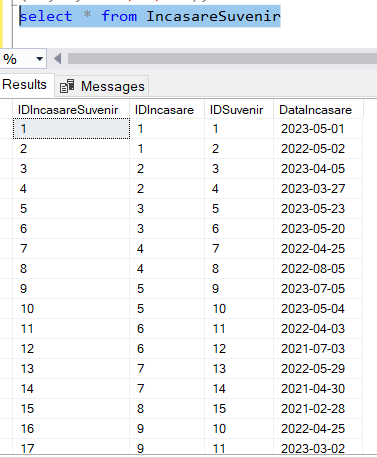
(7, 14,'2021/04/30'),

(8, 15,'2021/02/28'),

(9,10,'2022/04/25'),

(9,11,'2023/03/02'),

(10,10,'2022/04/05');

****

**14. Tabelul Clienti**

INSERT INTO Clienti (Nume, Prenume, ClientFidel, TipClient, Telefon, Mail, DataUltimaVizita)

VALUES

('Popescu', 'Ion', 'DA', 'VIP', '0712345678', 'ion.popescu@gmail.com', '2022-03-15'),

('Ionescu', 'Maria', 'NU', 'EXTENDED', '0756789123', 'maria.ionescu@yahoo.com', '2022-01-10'),

('Georgescu', 'Andrei', 'NU', 'REGULAR', '0723456789', 'andrei.georgescu@hotmail.com', '2022-04-01'),

('Dumitrescu', 'Cristina', 'NU', 'VIP', '0734567891', 'cristina.dumitrescu@gmail.com', '2022-02-20'),

('Popa', 'Gabriel', 'DA', 'REGULAR', '0745678912', 'gabriel.popa@gmail.com', '2022-04-10'),

('Radu', 'Diana', 'NU', 'EXTENDED', '0722222222', 'diana.radu@yahoo.com', '2022-03-01'),

('Munteanu', 'Alexandru', 'NU', 'VIP', '0733333333', 'alexandru.munteanu@hotmail.com', '2022-02-15'),

('Balan', 'Adrian', 'DA', 'REGULAR', '0755555555', 'adrian.balan@gmail.com', '2022-04-05'),

('Stefanescu', 'Cristina', 'NU', 'EXTENDED', '0711111111', 'cristina.stefanescu@yahoo.com', '2022-03-10'),

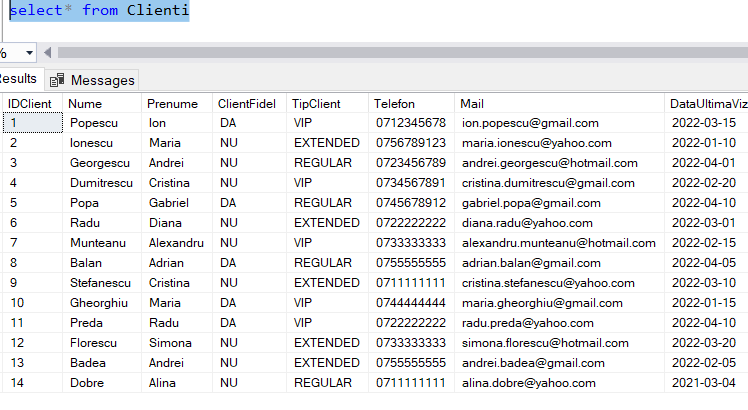
('Gheorghiu', 'Maria', 'DA', 'VIP', '0744444444', 'maria.gheorghiu@gmail.com', '2022-01-15'),

('Preda', 'Radu', 'DA', 'VIP', '0722222222', 'radu.preda@yahoo.com', '2022-04-10'),

('Florescu', 'Simona', 'NU', 'EXTENDED', '0733333333', 'simona.florescu@hotmail.com', '2022-03-20'),

('Badea', 'Andrei', 'NU', 'EXTENDED', '0755555555', 'andrei.badea@gmail.com', '2022-02-05'),

('Dobre', 'Alina', 'NU', 'REGULAR', '0711111111', 'alina.dobre@yahoo.com', '2021-03-04');

****

**15. Tabelul Expozitii**

INSERT INTO Expozitii (IDAngajat, Locatie, Descriere, CapacitatePersoane)

VALUES

(1, 'Galeria Renasterii', 'Expozitie de artă renascentistă', 50),

(2, 'Sala Impresionistă', 'Expozitie de artă impresionistă', 75),

(3, 'Sala Sculpturii', 'Expozitie de sculptură contemporană', 30),

(4, 'Sala de Artă Modernă', 'Expozitie de artă modernă', 40),

(5, 'Galeria Antică', 'Expozitie de artefacte antice', 60),

(6, 'Galeria de Arte Plastice', 'Expozitie de artă contemporană', 45),

(7, 'Sala Orientală', 'Expozitie de artă orientală', 25),

(8, 'Sala de Artă Clasică', 'Expozitie de artă clasică', 35),

(9, 'Sala de Arte Decorative', 'Expozitie de arte decorative', 50),

(10, 'Sala de Artă Fotografică', 'Expozitie de artă fotografică', 20),

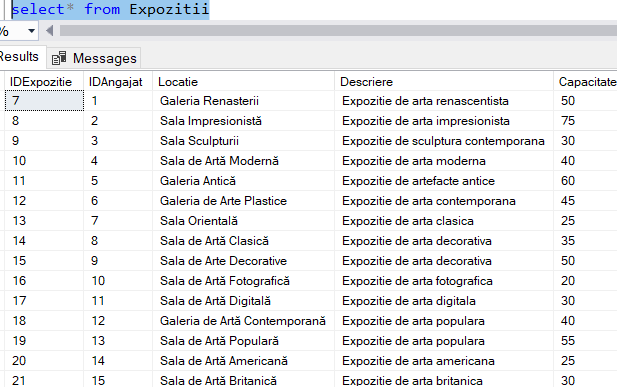
(11, 'Sala de Artă Digitală', 'Expozitie de artă digitală', 30),

(12, 'Galeria de Artă Contemporană', 'Expozitie de artă contemporană', 40),

(13, 'Sala de Artă Populară', 'Expozitie de artă populară', 55),

(14, 'Sala de Artă Americană', 'Expozitie de artă americană', 25),

(15, 'Sala de Artă Britanică', 'Expozitie de artă britanică', 30);

****

**16. Tabelul Rezervari**

insert into Rezervari(IDExpozitie, IDClient, NumarBilete, DataRezervare, OraRezervare)

values

(21, 3, 2, '2023-05-01', 10),

(7, 2, 3, '2023-05-02', 11),

(7, 5, 1, '2023-05-03', 12),

(9, 1, 4, '2023-05-04', 13),

(10, 4, 2, '2023-05-05', 14),

(11, 6, 1, '2023-05-06', 15),

(12, 7, 3, '2023-05-07', 16),

(12, 8, 2, '2023-05-08', 17),

(9, 9, 1, '2023-05-09', 18),

(7, 10, 2, '2023-05-10', 19),

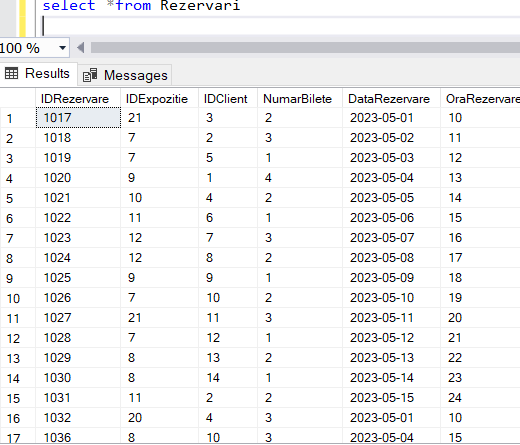
(21, 11, 3, '2023-05-11', 20),

(7, 12, 1, '2023-05-12', 21),

(8, 13, 2, '2023-05-13', 22),

(8, 14, 1, '2023-05-14', 23),

(11, 2, 2, '2023-05-15', 24)

****

**17. Tabelul Exponate**

INSERT INTO Exponate(IDExpozitie, DataAparitie, Nume, Tip, TaraAparitie, Descriere)

VALUES

(7, '2022-01-01', 'Tablou abstract', 'Pictură', 'Franța', 'O operă de artă abstractă'),

(7, '2022-01-01', 'Vază din porțelan', 'Sculptură', 'China', 'O vază frumoasă și delicată'),

(7, '2022-01-01', 'Statuie de bronz', 'Sculptură', 'Grecia', 'O statuie reprezentând zeul Zeus'),

(8, '2022-02-15', 'Colier din aur', 'Bijuterie', 'India', 'Un colier frumos din aur'),

(8, '2022-02-15', 'Brățară din argint', 'Bijuterie', 'Mexic', 'O brătară masivă din argint'),

(8, '2022-02-15', 'Inel cu diamant', 'Bijuterie', 'Belgia', 'Un inel cu un diamant mare și strălucitor'),

(9, '2022-03-10', 'Cana din ceramică', 'Ceramică', 'Spania', 'O cană frumoasă si rustică din ceramică'),

(10, '2022-03-10', 'Farfurie de porțelan', 'Ceramică', 'Japonia', 'O farfurie frumoasă si delicată din porțelan'),

(9, '2022-03-10', 'Vază din sticlă', 'Ceramică', 'Italia', 'O vază mare și transparentă din sticlă'),

(10, '2022-04-05', 'Broșă din perle', 'Bijuterie', 'Franța', 'O brosă elegantă din perle'),

(10, '2022-04-05', 'Cercei cu pietre semiprețioase', 'Bijuterie', 'Thailanda', 'Cercei lungi și strălucitori cu pietre semipretioase'),

(11, '2022-04-05', 'Brățară din aur', 'Bijuterie', 'Italia', 'O brătară subtire și elegantă din aur'),

(11, '2022-05-20', 'Vază din argint', 'Sculptură', 'Turcia', 'O vază masivă din argint'),

(11, '2022-05-20', 'Statuie de marmură', 'Sculptură', 'Grecia', 'O statuie frumoasă și realistă a lui Apollo'),

(12, '2022-05-20', 'Tablou în ulei', 'Pictură', 'Italia', 'Un tablou reprezentând o priveliste din Veneția'),

(12, '2022-06-11', 'Farfurie din ceramică', 'Ceramică', 'Turcia', 'O farfurie mică si rustică din ceramică'),

(12, '2022-06-11', 'Vază din porțelan', 'Sculptură', 'China', 'O vază frumoasă si delicată'),

(21, '2022-06-11', 'Tablou în ulei', 'Pictură', 'Spania', 'Un tablou impresionist reprezentând o grădină înflorită'),

(13, '2022-07-02', 'Broșă din aur', 'Bijuterie', 'Italia', 'O brosă subțire și elegantă din aur'),

(13, '2022-07-02', 'Cercei din argint', 'Bijuterie', 'Mexic', 'Cercei lungi si masivi din argint'),

(14, '2022-07-02', 'Inel cu rubin', 'Bijuterie', 'Thailanda', 'Un inel cu un rubin mare si frumos'),

(14, '2022-08-18', 'Statuie de marmură', 'Sculptură', 'Italia', 'O statuie frumoasă si detaliată a unei zeite'),

(14, '2022-08-18', 'Vază din argint', 'Sculptură', 'Turcia', 'O vază mare si masivă din argint'),

(15, '2022-08-18', 'Tablou abstract', 'Pictură', 'Franța', 'Un tablou abstract cu forme geometrice'),

(15, '2022-09-05', 'Farfurie de porțelan', 'Ceramică', 'Japonia', 'O farfurie frumoasă si delicată din porțelan'),

(16, '2022-09-05', 'Vază din sticlă', 'Ceramică', 'Italia', 'O vază mare și transparentă din sticlă'),

(16, '2022-09-05', 'Cana din ceramică', 'Ceramică', 'Spania', 'O cană frumoasă și rustică din ceramică'),

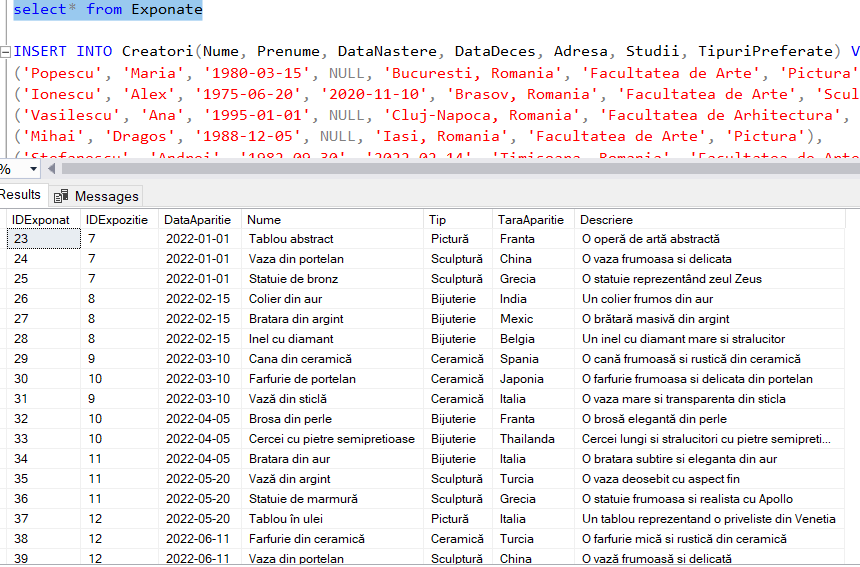
(17, '2022-10-22', 'Inel cu safir', 'Bijuterie', 'Belgia', 'Un inel cu un safir mare si frumos'),

(17, '2022-10-22', 'Cercei din aur', 'Bijuterie', 'Italia', 'Cercei mici si eleganti din aur'),

(18, '2022-10-22', 'Broșă din argint', 'Bijuterie', 'Grecia', 'O brosă masivă si rustică din argint'),

(19, '2022-11-11', 'Tablou în ulei', 'Pictură', 'Italia', 'Un tablou reprezentând o priveliste din Toscana'),

(20, '2022-11-11', 'Statuie de bronz', 'Sculptură', 'Grecia', 'O statuie frumoasă si realistă a lui Ares')

****

**18. Tabelul Creatori**

INSERT INTO Creatori(Nume, Prenume, DataNastere, DataDeces, Adresa, Studii, TipuriPreferate) VALUES

('Popescu', 'Maria', '1980-03-15', NULL, 'Bucuresti, Romania', 'Facultatea de Arte', 'Pictura'),

('Ionescu', 'Alex', '1975-06-20', '2020-11-10', 'Brasov, Romania', 'Facultatea de Arte', 'Sculptura'),

('Vasilescu', 'Ana', '1995-01-01', NULL, 'Cluj-Napoca, Romania', 'Facultatea de Arhitectura', 'Design de interior'),

('Mihai', 'Dragos', '1988-12-05', NULL, 'Iasi, Romania', 'Facultatea de Arte', 'Pictura'),

('Stefanescu', 'Andrei', '1982-09-30', '2022-02-14', 'Timisoara, Romania', 'Facultatea de Arte', 'Grafica'),

('Radu', 'Ioana', '1990-07-25', NULL, 'Constanta, Romania', 'Facultatea de Arte', 'Arta textila'),

('Popa', 'Andreea', '1985-04-10', NULL, 'Bacau, Romania', 'Facultatea de Arhitectura', 'Design de exterior'),

('Avram', 'Vlad', '1998-11-18', NULL, 'Pitesti, Romania', 'Facultatea de Arte', 'Fotografie'),

('Badea', 'Andrei', '1970-02-22', '2021-03-10', 'Suceava, Romania', 'Facultatea de Arte', 'Sculptura'),

('Pavel', 'Diana', '1993-08-08', NULL, 'Craiova, Romania', 'Facultatea de Arhitectura', 'Design de mobilier'),

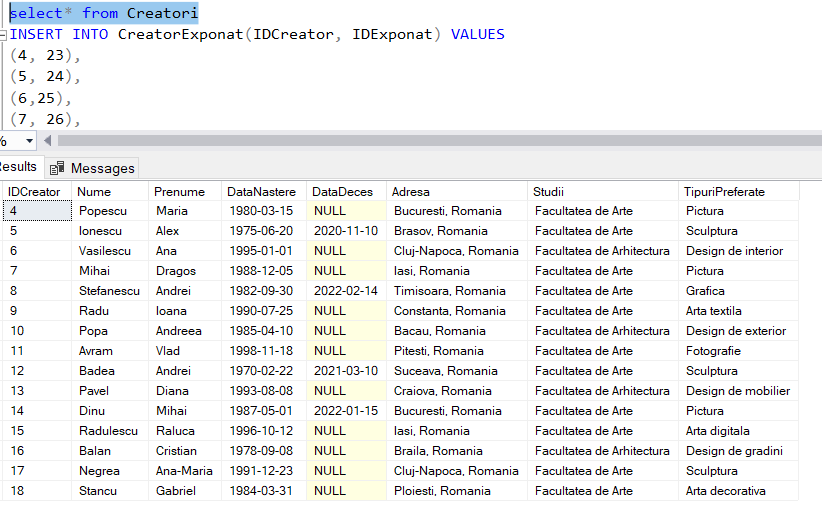
('Dinu', 'Mihai', '1987-05-01', '2022-01-15', 'Bucuresti, Romania', 'Facultatea de Arte', 'Pictura'),

('Radulescu', 'Raluca', '1996-10-12', NULL, 'Iasi, Romania', 'Facultatea de Arte', 'Arta digitala'),

('Balan', 'Cristian', '1978-09-08', NULL, 'Braila, Romania', 'Facultatea de Arhitectura', 'Design de gradini'),

('Negrea', 'Ana-Maria', '1991-12-23', NULL, 'Cluj-Napoca, Romania', 'Facultatea de Arte', 'Sculptura'),

('Stancu', 'Gabriel', '1984-03-31', NULL, 'Ploiesti, Romania', 'Facultatea de Arte', 'Arta decorativa');

****

**20. Tabelul CreatorExponat**

INSERT INTO CreatorExponat(IDCreator, IDExponat) VALUES

(4, 23),

(5, 24),

(6,25),

(7, 26),

(8, 27),

(9, 28),

(10, 29),

(11, 30),

(12, 31),

(13,32),

(14, 33),

(15, 34),

(12, 35),

(13, 36),

(15, 37),

(5,38),

(5,39),

(6,40),

(6,41),

(7,42),

(7,43),

(8,44),

(8,45),

(9,46),

(10,47),

(11,48),

(11,49),

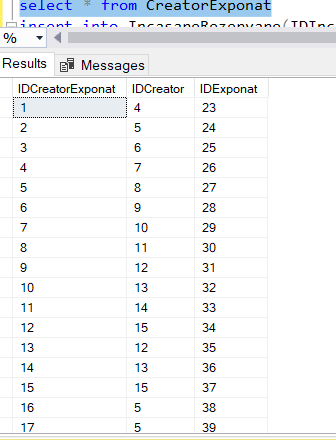
(12,50),

(13,51),

(12,52),

(12,53),

(13,54);

****

**21. Tabelul IncasareRezervare**

insert into IncasareRezervare(IDIncasare,IDRezervare) values

(1, 1017),

(2, 1019),

(3, 1020),

(4, 1018),

(5, 1021),

(6, 1022),

(7, 1023),

(8, 1024),

(9, 1025),

(10, 1026),

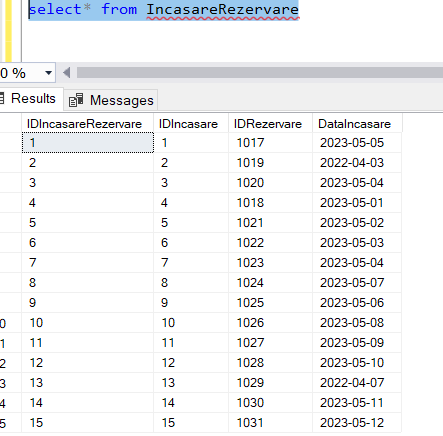
(11, 1027),

(12, 1028),

(13, 1029),

(14, 1030),

(15, 1031);

****

## 2.2. SELECT

**1. Sa se afiseze toate uneltele de tip „Electric”, cat si numele angajatilor care au acces la acestea.**

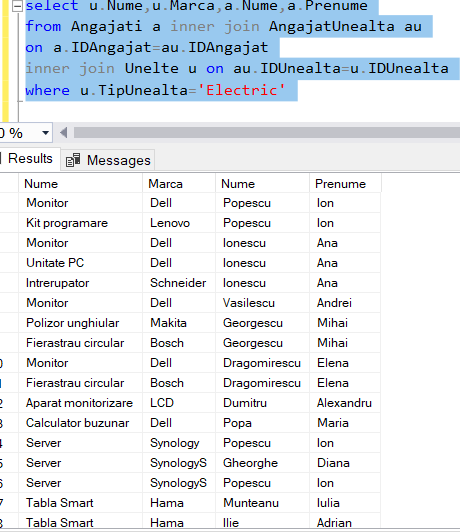
select u.Nume,u.Marca,a.Nume,a.Prenume

from Angajati a inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat

inner join Unelte u on au.IDUnealta=u.IDUnealta

where u.TipUnealta='Electric'



2. Sa se afiseze autorii operelor si numele operelor care sunt expuse in expozitia din „Galeria Renasterii”

select c.Nume as NumeAutor,c.Prenume as PrenumeAutor,e.Nume as NumeExponat

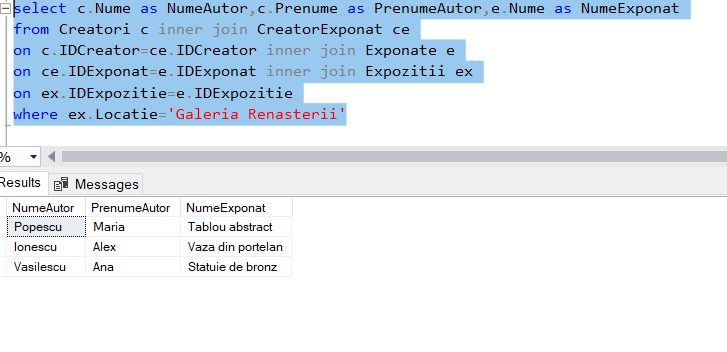
from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on ce.IDExponat=e.IDExponat inner join Expozitii ex

on ex.IDExpozitie=e.IDExpozitie

where ex.Locatie='Galeria Renasterii'



3. Sa se afiseze toti autorii operelor de arta(Nume,Prenume), operele acestora, respectiv locul unde are loc expozitia in care se afla exponatele lor, pentru autorii aflati inca in viata.

select c.Nume as NumeAutor,c.Prenume as PrenumeAutor,e.Nume as NumeExponate, ex.Locatie as LocatieExpozitie

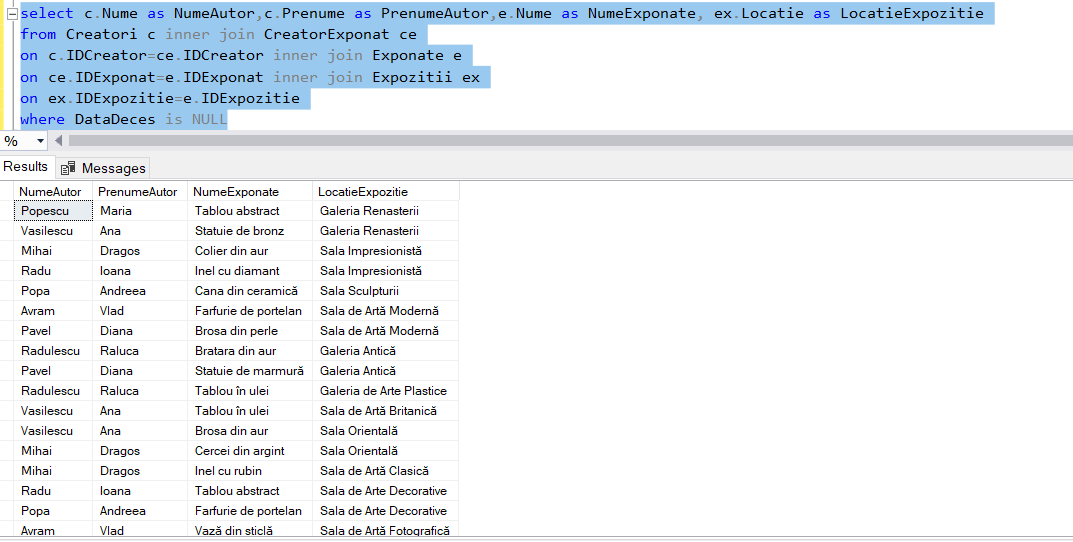
from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on ce.IDExponat=e.IDExponat inner join Expozitii ex

on ex.IDExpozitie=e.IDExpozitie

where DataDeces is NULL



4. Sa se afiseze toate incasarile obtinute in urma rezervarii clientilor al caror nume incepe cu P, cat si data la care s-a efectuat incasarea, ordonate descrescator dupa suma incasata.

select c.Nume,inc.SumaIncasata,ir.DataIncasare

from Clienti c inner join Rezervari r

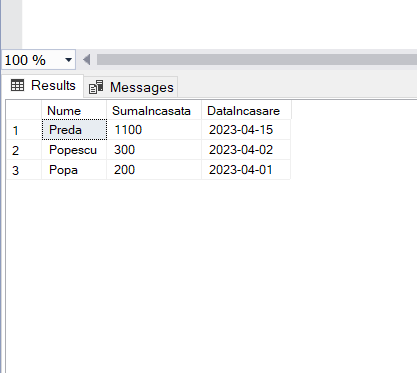
on c.IDClient=r.IDClient inner join IncasareRezervare ir

on ir.IDRezervare=r.IDRezervare inner join Incasari inc

on inc.IDIncasare=ir.IDIncasare

where c.Nume like 'P%'

order by inc.SumaIncasata desc



5.Sa se afiseze toti angajatii al caror procent de asigurare medicala depaseste

10%.

select a.Nume,a.Prenume,i.ProcentAsigurareMedicala

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat

inner join Functii f

on fa.IDFunctie=f.IDFunctie

inner join SalariuFunctie sa

on sa.IDFunctie=f.IDFunctie

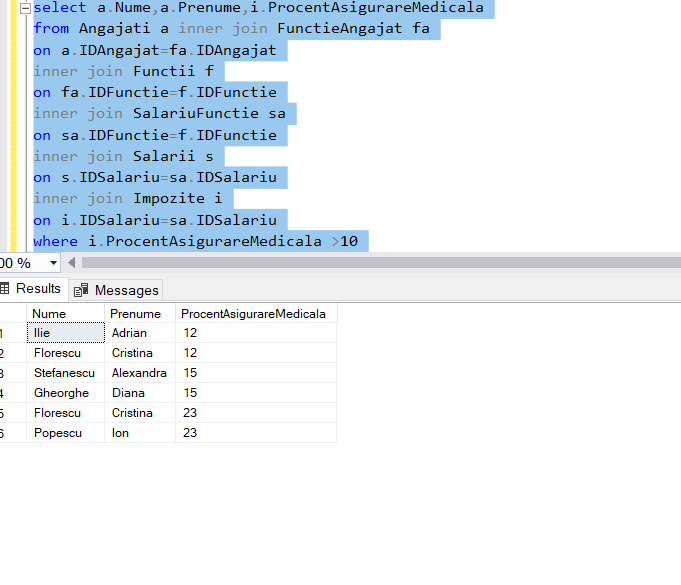
inner join Salarii s

on s.IDSalariu=sa.IDSalariu

inner join Impozite i

on i.IDSalariu=sa.IDSalariu

where i.ProcentAsigurareMedicala >10



6. Sa se afiseze top 3 furnizori ale caror suveniruri au depasit incasari de 699 de lei.

select top 3 f.NumeFurnizor,inc.SumaIncasata

from Furnizori f inner join FurnizorSuvenir fz

on f.IDFurnizor=fz.IDFurnizor inner join Suveniruri s

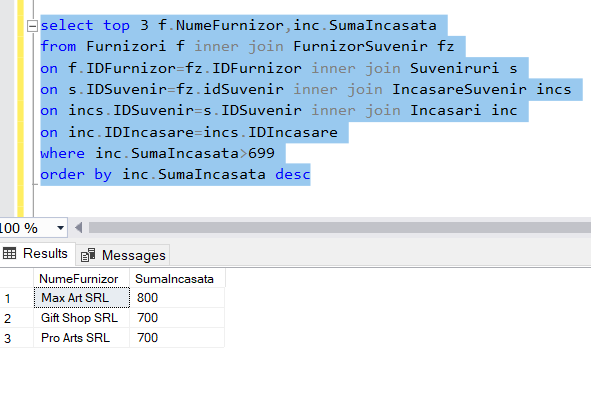
on s.IDSuvenir=fz.idSuvenir inner join IncasareSuvenir incs

on incs.IDSuvenir=s.IDSuvenir inner join Incasari inc

on inc.IDIncasare=incs.IDIncasare

where inc.SumaIncasata>699

order by inc.SumaIncasata desc



7. Sa se afiseze doar numele furnizorilor uneltelor care apar in inventarul angajatilor ce au salarii net peste 2000 lei.

select f.NumeFurnizor

from Furnizori f inner join FurnizorUnealta fu

on f.IDFurnizor=fu.IDFurnizor

inner join Unelte u

on fu.IDUnealta=u.IDUnealta

inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta

inner join Angajati a

on a.IDAngajat=au.IDAngajat

inner join FunctieAngajat fa

on fa.IDAngajat=a.IDAngajat

inner join Functii func

on fa.IDFunctie=func.IDFunctie

inner join SalariuFunctie sf

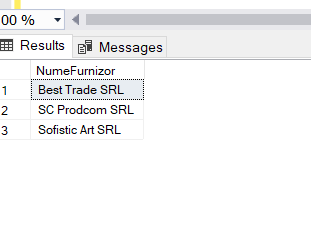
on sf.IDFunctie=func.IDFunctie

inner join Salarii sal

on sal.IDSalariu=sf.IDSalariu

where sal.SalariuNet>=2000

group by f.NumeFurnizor



8. Afisarea tuturor clientilor al caror e-mail se termina cu extensia gmail.com si care participa la o expozitie al carei angajat responsabil este angajatul cu id-ul numarul 1.

select c.IDClient,c.Nume,c.Prenume

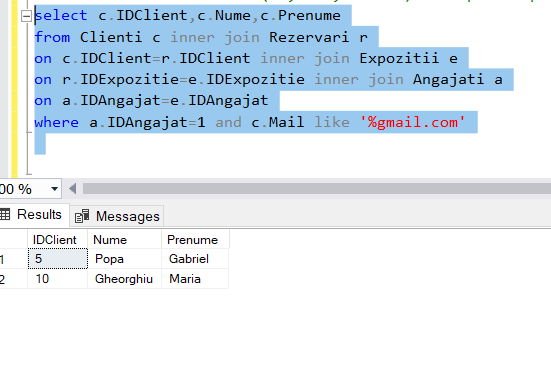
from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient inner join Expozitii e

on r.IDExpozitie=e.IDExpozitie inner join Angajati a

on a.IDAngajat=e.IDAngajat

where a.IDAngajat=1 and c.Mail like '%gmail.com'



9. Afisarea tuturor angajatilor(si functiilor acestora ) care au salarii(se afiseaza si acestea,brut si net) in care sunt incluse peste 5 ore suplimentare, iar data avansului este pe ziua de 1 a lunii, sortat in ordine descrescatoare a salariilor.

select a.Nume,f.NumeFunctie,s.SalariuBrut,s.SalariuNet

from Angajati a inner join FunctieAngajat fa

on fa.IDAngajat=a.IDAngajat inner join Functii f

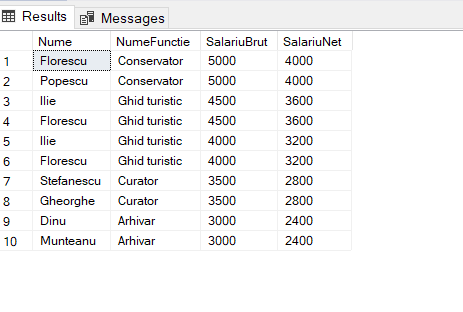
on f.IDFunctie=fa.IDFunctie inner join SalariuFunctie sf

on sf.IDFunctie=f.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where s.OreSuplimentare>5 and DAY(s.DataAvans)=1

order by s.SalariuBrut desc



10. Afisarea furnizorului(NumeFurnizor,ID,An de activitate pe piata) care asigura productia si livrarea de unelte pentru angajatul responsabil pe „Sala sculpturii” in cadrul expozitiilor.

select f.IDFurnizor as ID,f.NumeFurnizor as Nume, f.AnActivitatePiata as AnFurnizor

from Furnizori f inner join FurnizorUnealta fu

on f.IDFurnizor=fu.IDFurnizor inner join Unelte u

on u.IDUnealta=fu.IDUnealta inner join AngajatUnealta au

on au.IDUnealta=u.IDUnealta inner join Angajati a

on a.IDAngajat=au.IDAngajat inner join Expozitii e

on e.IDAngajat=a.IDAngajat

where e.Locatie='Sala sculpturii'

11. Afisarea tuturor creatorilor(Nume,Prenume,IDCreator) care au furnizat opere pentru angajatii care au in responsabilitatea lor „Tabla portabila”

select c.Nume,c.Prenume,c.IDCreator

from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on ce.IDExponat =e.IDExponat inner join Expozitii expo

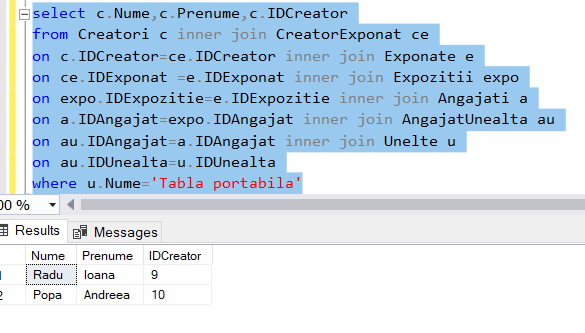
on expo.IDExpozitie=e.IDExpozitie inner join Angajati a

on a.IDAngajat=expo.IDAngajat inner join AngajatUnealta au

on au.IDAngajat=a.IDAngajat inner join Unelte u

on au.IDUnealta=u.IDUnealta

where u.Nume='Tabla portabila'



12. Afisarea tuturor angajatilor(IDAngajat,Nume,Prenume) care au diferenta dintre salariul brut si salariul net mai mare de 900 de lei si al caror numar de telefon incepe cu prefixul „072” . Se vor afisa si duplicate pentru angajatii care au mai multe functii.

select a.Nume,a.Prenume,a.NumarTelefon,f.NumeFunctie,s.SalariuBrut-s.SalariuNet as DiferentaSalarii

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu

where s.SalariuBrut-s.SalariuNet>900 and a.NumarTelefon like '072%'



13. Afisarea tuturor angajatilor(ID,Nume,Prenume) responsabili pe salile in care s-au facut rezervari care au adus incasari de peste 1000 lei.

select a.IDAngajat,a.Nume,a.Prenume,i.SumaIncasata

from Angajati a inner join Expozitii e

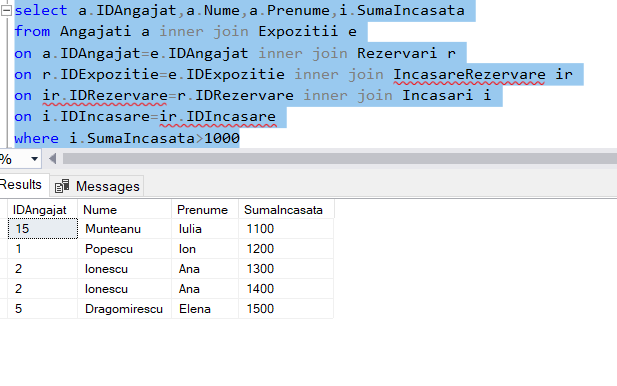
on a.IDAngajat=e.IDAngajat inner join Rezervari r

on r.IDExpozitie=e.IDExpozitie inner join IncasareRezervare ir

on ir.IDRezervare=r.IDRezervare inner join Incasari i

on i.IDIncasare=ir.IDIncasare

where i.SumaIncasata>1000



14. Afisarea tuturor clientilor(IDClient,Nume,Prenume) cu statutul VIP care iau contact cu salile de expozitie ale caror angajati fac parte din corpul de executie din cadrul muzeului.

select c.IDClient,c.Nume,c.Prenume

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient inner join Expozitii e

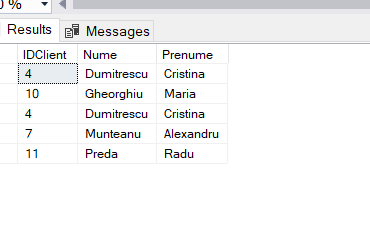
on e.IDExpozitie=r.IDExpozitie inner join Angajati a

on a.IDAngajat=e.IDAngajat inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

where f.Executie='DA' and c.TipClient='VIP'



15. Afisarea tuturor angajatilor care au in grija sali pentru care exista exponate realizate de creatori din „Bucuresti, Romania”

select a.IDAngajat,a.Nume,a.Prenume

from Angajati a inner join Expozitii e

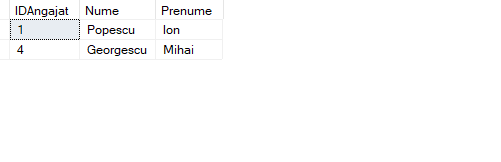
on a.IDAngajat=e.IDAngajat inner join Exponate ex

on e.IDExpozitie=ex.IDExpozitie inner join CreatorExponat ce

on ex.IDExponat=ce.IDExponat inner join Creatori c

on ce.IDCreator=c.IDCreator

where c.Adresa='Bucuresti, Romania'



16. Afisarea tuturor functiilor(Idfunctie + NumeFunctie) pentru care suma totala a impozitelor prezentate in baza de date depaseste 25%

select f.IDFunctie,f.NumeFunctie

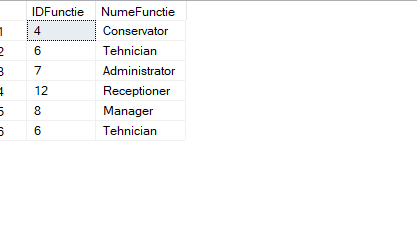
from Functii f inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu inner join Impozite i

on s.IDSalariu=i.IDSalariu

where i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat>25



17. Afisarea angajatilor, a salariilor tuturor angajatilor din baza de date, a functiilor acestora si a salariilor obtinute in urma micsorarii de pe urma impozitelor trecute in baza de date( SalariuBrut-SalariuBrut\*Impozite/100)

select a.IDAngajat,a.Nume,a.Prenume,f.NumeFunctie, s.SalariuBrut-s.SalariuBrut\*(i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat)/100 as Salariu

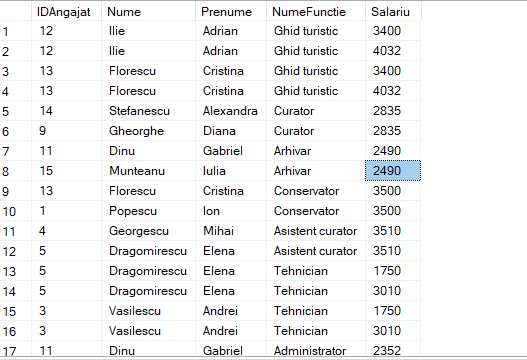
from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu inner join Impozite i



18. Afisarea tuturor exponatelor care se afla in expozitia pentru care sunt responsabili angajati cu functia de tehnicieni si programul orar „9:00 – 17:00”

select e.IDExponat,e.Nume,e.Descriere

from Exponate e inner join Expozitii expo

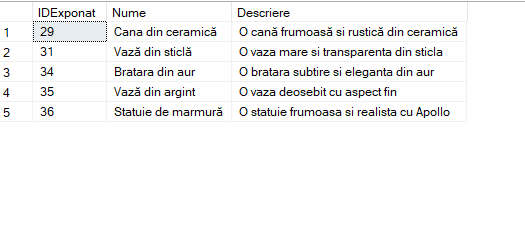
on e.IDExpozitie=expo.IDExpozitie inner join Angajati a

on expo.IDAngajat=a.IDAngajat inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

where f.NumeFunctie='Tehnician' and f.IntervalOrar='9:00 - 17:00'



19. Afisarea tuturor clientilor care au avut o rezervare dupa data in care a rezervat clientul cu numele Balan Adrian.

select c.IDClient,c.Nume,c.Prenume,r.DataRezervare

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

where r.DataRezervare> ALL

(

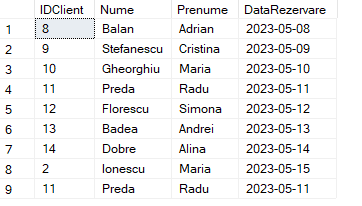
select r.DataRezervare

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

where c.Nume='Balan' and c.Prenume='Adrian'

)



20. Afisarea tuturor angajatilor care sunt mai vechi in muzeu decat angajatul administrator sau angajatul Manager, se va face alegerea in functie de cel mai nou dintre cei doi angajati.

select a.IDAngajat,a.Nume,a.Prenume

from Angajati a

where a.DataAngajare<

(Select top (1) a.DataAngajare

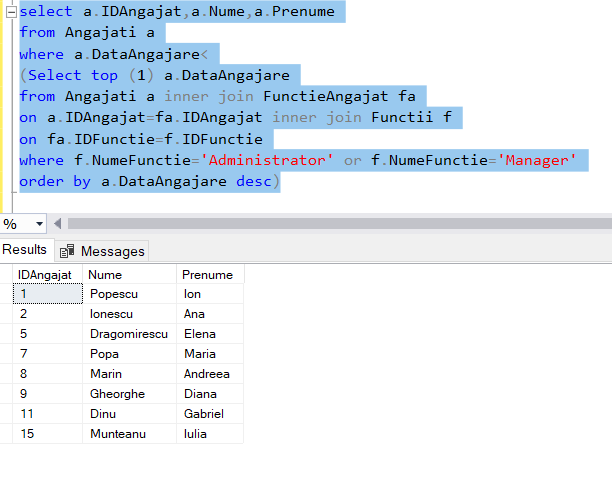
from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

where f.NumeFunctie='Administrator' or f.NumeFunctie='Manager'

order by a.DataAngajare desc)



21. Selectarea tuturor angajatilor care au un salariu mai mare decat cel al tehnicianului cu cel mai mare salariu.

select distinct a.Nume,a.Prenume

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu

where s.SalariuNet>

(select top(1) s.SalariuNet

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

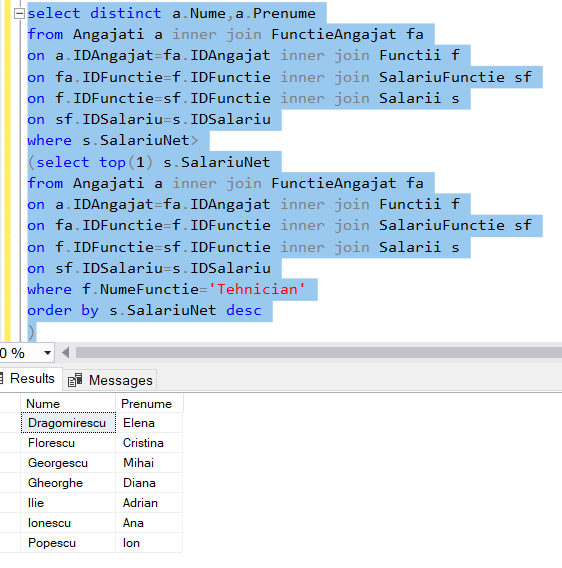
on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu

where f.NumeFunctie='Tehnician'

order by s.SalariuNet desc

)



22. Afisarea codului SRL al tuturor furnizorilor cu durata contractului de furnizare de cel putin 12 ani care genereaza incasari prin produsele acestora de peste 500 de lei.

select f.IDFurnizor,f.codSRL

from Furnizori f inner join FurnizorSuvenir fs

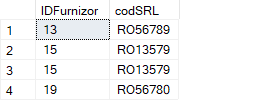
on f.IDFurnizor=fs.IDFurnizor inner join Suveniruri s

on fs.IDSuvenir=s.IDSuvenir inner join IncasareSuvenir ins

on s.IDSuvenir=ins.IDsuvenir inner join Incasari inc

on ins.IDIncasare=inc.IDIncasare

where inc.SumaIncasata>500 and f.DurataContractFurnizare>=12



23. Afisarea tuturor exponatelor(ID,Nume,Descriere) care se afla in expozitiile angajatilor din Bucuresti

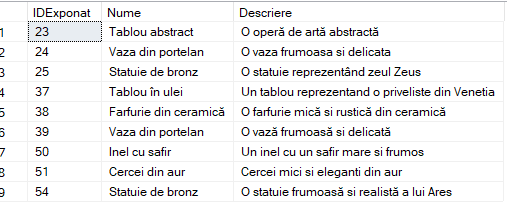
select expo.IDExponat,expo.Nume,expo.Descriere

from Angajati a inner join Expozitii e

on a.IDAngajat=e.IDAngajat inner join Exponate expo

on e.IDExpozitie=expo.IDExpozitie

where a.Localitate='Bucuresti'



24. Afisarea tuturor clientilor care au avut data ultimei vizite inainte de angajarea lui Vasilescu Andrei.

select IDClient,Nume,Prenume,DataUltimaVizita

from Clienti

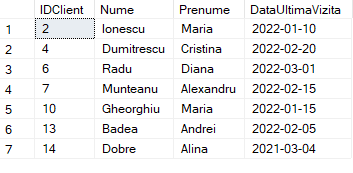
where DataUltimaVizita <(

select DataAngajare

from Angajati

where Nume='Vasilescu' and Prenume='Andrei'

)



25. Afisarea tuturor incasarilor mai mari decat valoarea maxima a avansului unui manager.

select IDIncasare,SumaIncasata

from Incasari

where Incasari.SumaIncasata>

(select top(1) s.ValoareAvans

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

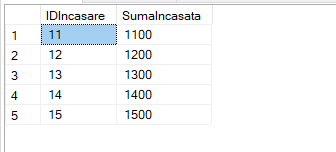
on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where f.NumeFunctie='Manager'

order by s.ValoareAvans desc)



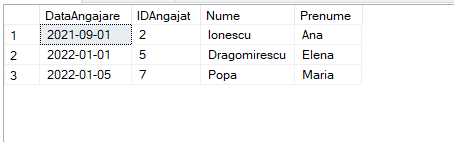
**26. Afisarea celor mai vechi 3 angajati din cadrul muzeului.**

select top(3) DataAngajare as DataAngajare,IDAngajat,Nume,Prenume

from Angajati

group by DataAngajare,IDAngajat,Nume,Prenume

order by DataAngajare

****

**27. Afisarea angajatilor(ID,Nume,Prenume) care au in posesie mai mult de 3 unelte si numarul acestora.**

select a.IDAngajat,a.Nume,a.Prenume,count(\*) as NumarUnelte

from Angajati a inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat inner join Unelte u

on au.IDUnealta=u.IDUnealta

group by a.IDAngajat,a.Nume,a.Prenume

having count(\*)>3



**28. Afisarea gruparii angajatilor in functie de impozitul total platit conform bazei de date in care angajatii au un impozit mai mare de 5 la suta pentru Casa de Pensii.**

select i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat as ImpozitTotal,a.IDAngajat,a.Nume,a.Prenume

from Angajati a inner join FunctieAngajat fs

on a.IDAngajat=fs.IDAngajat inner join Functii f

on fs.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu inner join Impozite i

on s.IDSalariu=i.IDSalariu

group by i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat,i.ProcentCasaPensii,a.IDAngajat,a.Nume,a.Prenume

having i.ProcentCasaPensii>5

****

**29. Afisarea clientilor de tip VIP care participa la expozitii in care capacitatea este mai mica de 75 de persoane.**

select c.Nume,c.Prenume,e.CapacitatePersoane

from Clienti c inner join Rezervari r

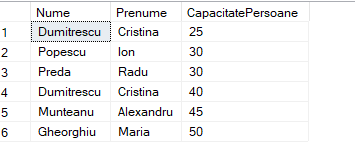
on c.IDClient=r.IDClient inner join Expozitii e

on r.IDExpozitie=e.IDExpozitie

where c.TipClient='VIP'

group by e.CapacitatePersoane,c.Nume,c.Prenume

having e.CapacitatePersoane<75

****

**30. Afisarea adresei furnizorilor care produc doar 2 produse de tip suvenir.**

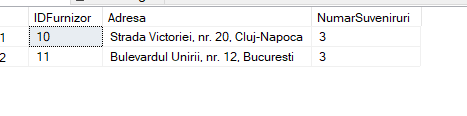
select f.IDFurnizor,f.Adresa,count(\*) as NumarSuveniruri

from Furnizori f inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor

group by f.IDFurnizor,f.Adresa

having count(\*)=3

****

**31. Afisarea codului SRL al furnizorilor care produc numarul minim de unelte.**

select f.IDFurnizor,f.CodSRL

from Furnizori f inner join FurnizorUnealta fu

on f.IDFurnizor=fu.IDFurnizor inner join Unelte u

on fu.IDUNealta=u.IDUnealta

group by f.IDFurnizor,f.CodSRL

having count(\*)=

(select top(1) count(\*)

from Furnizori f inner join FurnizorUnealta fu

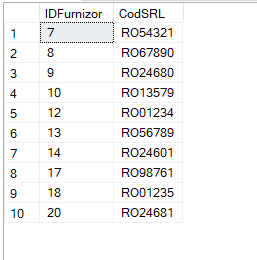
on f.IDFurnizor=fu.IDFurnizor inner join Unelte u

on fu.IDUnealta=u.IDUnealta

group by f.IDFurnizor

order by count(\*)

)



32. Afisarea tuturor angajatilor grupati in functie de numarul de persoane trecute in rezervari pentru salile pe care sunt responsabili, fiind obligatorie depasirea numarului de 2 bilete achitate

select a.IDAngajat,a.Nume,a.Prenume

from Angajati a inner join Expozitii e

on a.IDAngajat=e.IDAngajat inner join Rezervari r

on e.IDExpozitie=r.IDExpozitie inner join Clienti c

on r.IDClient=c.IDClient

group by r.NumarBilete,a.IDAngajat,a.Nume,a.Prenume

having r.NumarBilete=2



33. Afisarea salariilor (Brut,Net) si a angajatilor (Nume,Prenume,AniExperienta) care corespund acestor salarii.

select s.SalariuBrut,s.SalariuNet,a.Nume,a.Prenume,a.AniExperientaDomeniu

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu

group by s.SalariuBrut,s.SalariuNet,a.Nume,a.Prenume,a.AniExperientaDomeniu



34. Afisarea uneltelor si a numarului de angajati care au acces la acestea.

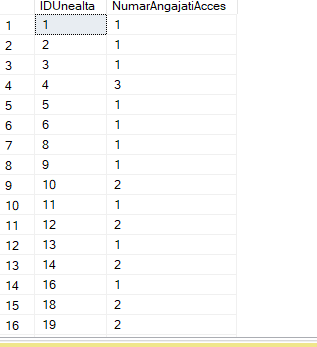
select u.IDUnealta,count(au.IDUnealta) as NumarAngajatiAcces

from Unelte u inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta inner join Angajati a

on au.IDAngajat=a.IDAngajat

group by u.IDUnealta



35. Afisarea tuturor salilor si a descrierii acestora , a numarului de bilete rezervate de clientii pentru acestea, a numarului de telefon al clientilor care au facut o rezervare pentru o ora mai tarzie de ora 15.

select e.Locatie,e.Descriere,r.NumarBilete,c.Telefon

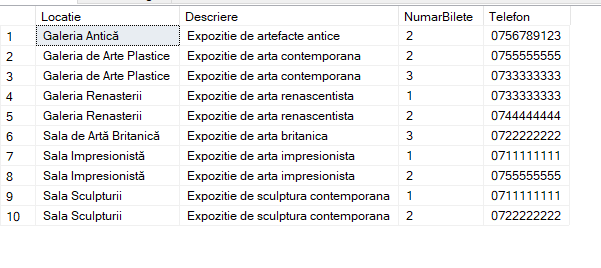
from Expozitii e inner join Rezervari r

on e.IDExpozitie=r.IDExpozitie inner join Clienti c

on r.IDClient=c.IDClient

where r.OraRezervare>15

group by e.Locatie,e.Descriere,r.NumarBilete,c.Telefon



36. Afisarea furnizorilor, a numarului de unelte furnizate pentru angajatii care se afla in departamentul de Ingrijire.

select f.NumeFurnizor,f.CodSRL,count(fu.IDUnealta) as NumarUnelte

from Furnizori f inner join FurnizorUnealta fu

on f.IDFurnizor=fu.IDFurnizor inner join Unelte u

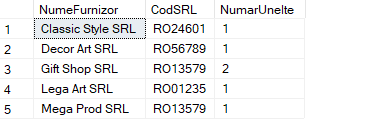
on fu.IDUnealta=u.IDUnealta inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta inner join Angajati a

on au.IDAngajat=a.IDAngajat

where a.Departament='Ingrijire'

group by f.NumeFurnizor,f.CodSRL,fu.IDUnealta



37. Afisarea tuturor furnizorilor si a numarului de suveniruri furnizate cu tara de provenienta fiind tara suvenirului cu id-ul 7 pentru care s-au incasat sub 1000 de lei.

select f.NumeFurnizor,count(fs.IDFurnizor) as NumarSuveniruriCerute

from Furnizori f inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor inner join Suveniruri s

on fs.IDSuvenir=s.IDSuvenir inner join IncasareSuvenir incs

on s.IDSuvenir=incs.IDSuvenir inner join Incasari inc

on incs.IDIncasare=inc.IDIncasare

where inc.SumaIncasata<1500 and s.TaraProvenienta =

(

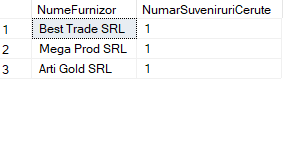
select TaraProvenienta

from Suveniruri

where IDSuvenir=1

)

group by f.NumeFurnizor,fs.IDFurnizor



38. Afisarea tuturor uneltelor care au ca numar de utilizatori, numarul maxim de utilizatori posibil pe care-l poate avea o unealta.

select max(u.Nume) as NumeUnealta ,max(u.Marca) as Marca,count(au.IDAngajat) as NumarUtilizatori

from Unelte u inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta inner join Angajati a

on au.IDAngajat=a.IDAngajat

group by u.IDUnealta

having count(au.IDAngajat)=

(

select top(1) count(au.IDAngajat)

from Unelte u inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta inner join Angajati a

on au.IDAngajat=a.IDAngajat

group by u.IDUnealta

order by count(au.IDAngajat) desc

)



39. Afisarea tuturor uneltelor care se afla in grija oamenilor din Departamentul „Ingrijire” si persoanele responsabile de acestea.

select u.IDUnealta,u.Nume,a.Nume,a.Prenume,a.IDAngajat

from Unelte u inner join AngajatUnealta au

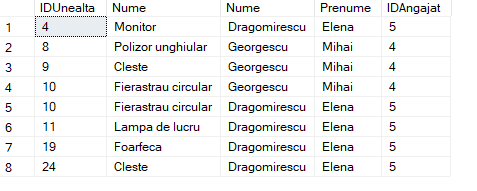
on u.IDUnealta=au.IDUnealta inner join Angajati a

on au.IDAngajat=a.IDAngajat inner join FunctieAngajat fa

on a.IDAngajat= fa.IDAngajat

where a.Departament='Ingrijire'

group by u.IDUnealta,u.Nume,a.Nume,a.Prenume,a.IdAngajat



40. Afisarea unei statistici care sa cuprinda anul scoaterii la concurs a unei functii si numarul de angajati care indeplinesc acea functie.

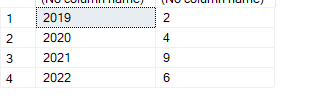
select YEAR(f.DataScoatereLaConcurs),count(fa.IDAngajat)

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

group by YEAR(f.DataScoatereLaConcurs)



41. Afisarea angajatului care are cel mai mare numar de clienti prezenti la expozitia acestuia.

select top(1) a.Nume,a.Prenume,count(c.IDClient) as NumarClienti

from Angajati a inner join Expozitii e

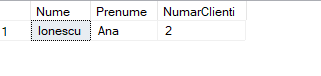
on a.IDAngajat=e.IDAngajat inner join Rezervari r

on r.IDExpozitie=e.IDExpozitie inner join Clienti c

on c.IDClient=r.IDClient

group by a.Nume,a.Prenume,c.IDClient

order by count(c.IDClient) desc



42. Afisarea salariului net mediu pentru persoanele din cadrul departamentelor de Ingrijire si Management.

select sum(s.SalariuNet)/count(s.SalariuNet) as SalariuMediu

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where a.Departament in ('Ingrijire','Management')



43. Afisarea sumei pe care muzeul o poate obtine din vanzarea stocurilor de tricouri actuale.

select sum(Pret\*BucatiStoc) as SumaIncasata

from Suveniruri

where Nume like 'Tricou%'



44. Afisarea unei statistici care sa cuprinda numele unui creator si numarul de opere expuse in muzeu create de catre acesta, unde creatorii sunt absolventi ai facultatii de arta.

select c.IDCreator,count(\*) as NumarCreatiiExpuse

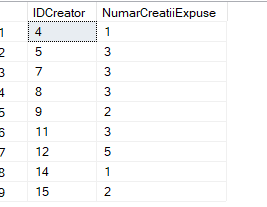
from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on ce.IDExponat=e.IDExponat

where c.Studii='Facultatea de Arte'

group by c.IDCreator



45. Afisarea tuturor detaliilor despre furnizori cu exceptia celor al caror cod SRL se termina cu cifra 0.

select \*

from Furnizori

EXCEPT

select \*

from Furnizori

where CodSRL like '%0'



46. Afisarea tuturor furnizorilor din orasele Bucuresti sau Cluj-Napoca.

select\*

from Furnizori

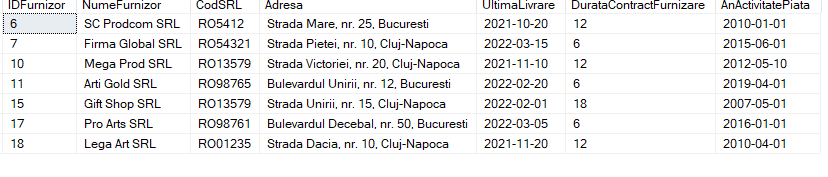
where Adresa like '%Bucuresti'

Union

select\*

from Furnizori

where Adresa like '%Cluj-Napoca'



47. Afisarea tuturor suvenirurilor al caror id este un numar divizibil cu 5 si cu 2.

select\*

from Suveniruri

where IDSuvenir%2=0

intersect

select\*

from Suveniruri

where IDSuvenir%5=0



48. Afisarea angajatilor care au functie de conducere si au mai mult de 2 ani experienta in domeniu.

select a.IDAngajat,a.Nume,a.Prenume

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

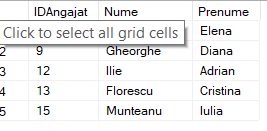
where f.Executie='DA'

intersect

select IDAngajat,Nume,Prenume

from Angajati

where AniExperientaDomeniu>2



49. Afisarea tuturor clientilor(IDClient,NumeClient,PrenumeClient,NumarTelefon) care au conturi de gmail si nu au cont tip REGULAR.

select IDClient,Nume,Prenume,Telefon,TipClient

from Clienti

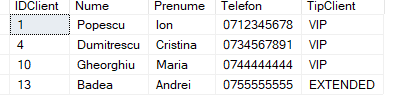
where Mail like '%@gmail.com'

EXCEPT

select IDClient,Nume,Prenume,Telefon,TipClient

from Clienti

where Mail like '%@gmail.com' and TipClient = 'REGULAR'



50. Afisarea tuturor expozitiilor care contin bijuterii si au un numar de cel putin 2 prezente in expunerea acestora.

select e.IDExpozitie,e.Locatie,count(e.IDExpozitie) as NumarVaze

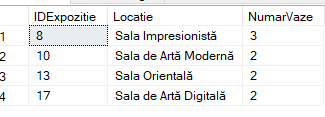
from Expozitii e inner join Exponate ex

on e.IDExpozitie=ex.IDExpozitie

where ex.Tip='Bijuterie'

group by e.IDExpozitie,e.Locatie

having count(e.IDExpozitie)>=2



51. Afisarea creatorilor/creatorului care se afla la originea unui exponat din expozitia angajatului ale carui unelte sunt furnizate de catre furnizorii al caror cod SRL se termina cu cifra 0.

select distinct c.IDCreator,c.Nume,c.Prenume

from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate expo

on ce.IDExponat=expo.IDExponat inner join Expozitii e

on expo.IDExpozitie=e.IDExpozitie inner join Angajati a

on e.IDAngajat=a.IDAngajat inner join AngajatUnealta au

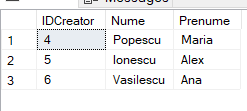
on a.IDAngajat=au.IDAngajat inner join Unelte u

on au.IDUnealta=u.IDUnealta inner join FurnizorUnealta fu

on u.IDUnealta=fu.IDUnealta inner join Furnizori f

on fu.IDFurnizor=f.IDFurnizor

where f.CodSRL like '%0'



52. Afisarea numarului de angajati care au salarii net stocate in baza de date avand indecsi impari.

select s.SalariuNet,count(\*) as NumarAngajati

from Angajati a inner join FunctieAngajat fu

on a.IDAngajat=fu.IDAngajat inner join Functii f

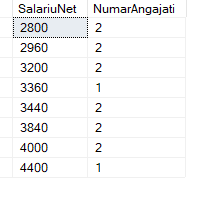
on fu.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where s.IDSalariu % 2 =1

group by s.SalariuNet



53. Afisarea incasarilor totale provenite din Magneti.

select sum(i.SumaIncasata) as SumaTotala

from Suveniruri s inner join IncasareSuvenir incs

on s.IDSuvenir=incs.IDSuvenir inner join Incasari i

on incs.IDIncasare=i.IDincasare

where s.Nume='Magnet'



54. Afisarea numarului de angajati ai firmei care au impozitul de 10 la suta pentru asigurarea medicala.

declare @variabila int=10

select count(\*) as NumarAngajati

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

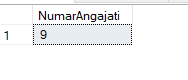
on f.IDFunctie=sf.IDFunctie inner join Salarii s

on sf.IDSalariu=s.IDSalariu inner join Impozite i

on s.IDSalariu=i.IDSalariu

where i.ProcentAsigurareMedicala=@variabila

group by i.ProcentAsigurareMedicala



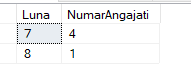
55. Afisarea numarului de angajati carora le expira contractul pe timpul verii.

select MONTH(a.DataExpirareContract) as Luna ,count(\*) as NumarAngajati

from Angajati a

where MONTH(a.DataExpirareContract) in ('6','7','8')

group by MONTH(a.DataExpirareContract)



56. Afisarea tuturor angajatilor care au avut prezenti clienti in sala acestora cu numarul de telefon care se termina intr-o cifra impara.

select a.IDAngajat,a.Nume,a.Prenume

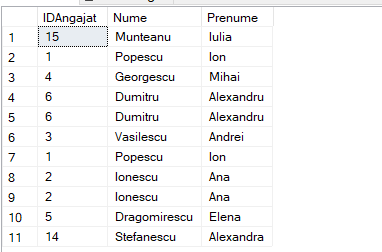
from Angajati a inner join Expozitii e

on a.IDAngajat=e.IDAngajat inner join Rezervari r

on r.IDExpozitie=e.IDExpozitie inner join Clienti c

on c.IDClient=r.IDClient

where c.Telefon like '%1' or c.Telefon like '%3' or c.Telefon like '%5' or c.Telefon like '%7' or c.Telefon like '%9'



57. Afisarea numarului de angajati pentru furnizorii de suveniruri al caror codSRL se termina cu cifra 0.

select f.NumeFurnizor,count(\*) as NumarAngajati

from Angajati a inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat inner join Unelte u

on au.IDUnealta=u.IDUnealta inner join FurnizorUnealta fu

on u.IDUnealta= fu.IDUnealta inner join Furnizori f

on fu.IDFurnizor=f.IDFurnizor

where f.CodSRL like '%0'

group by f.NumeFurnizor



58. Afisarea tuturor angajatilor care au in grija exponate(se va afisa si numele acestora+ descriere) din Grecia sau Italia.

select a.IDAngajat,a.Nume,a.Prenume,a.NumarTelefon,expo.Nume,expo.Descriere

from Angajati a inner join Expozitii e

on a.IDAngajat=e.IDAngajat inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

where expo.TaraAparitie='Grecia'

union

select a.IDAngajat,a.Nume,a.Prenume,a.NumarTelefon,expo.Nume,expo.Descriere

from Angajati a inner join Expozitii e

on a.IDAngajat=e.IDAngajat inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

where expo.TaraAparitie='Italia'

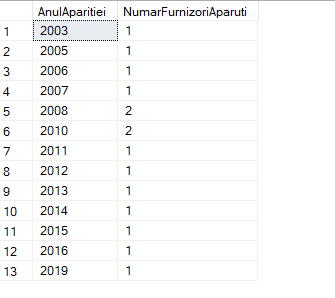


59. Afisarea unei statistici a furnizorilor ce consta in gruparea in functie de anul de aparitie a numarului de furnizori infiintati in acel an.

select YEAR(AnActivitatePiata) as AnulAparitiei,count(\*) as NumarFurnizoriAparuti

from Furnizori

group by YEAR(AnActivitatePiata)



60. Afisarea tuturor clientilor care nu sunt fideli si iau contact cu angajati care ocupa functia de arhivar.

select c.IDClient,c.Nume,c.Prenume,a.Nume,a.Prenume

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient inner join Expozitii ex

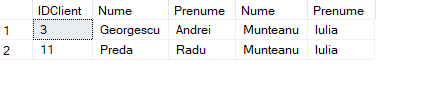
on r.IDExpozitie=ex.IDExpozitie inner join Angajati a

on ex.IDAngajat=a.IDAngajat inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie

where f.NumeFunctie='Arhivar'



61. Afisarea clientului care are cele mai multe rezervari si numarul acestora.

select c.Nume,c.Prenume,count(\*) as NumarRezervari

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

group by c.Nume,c.Prenume

having count(\*)=( select top(1) count(\*) as NumarRezervari

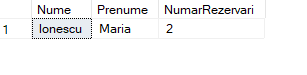
from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

group by c.IDClient

order by NumarRezervari desc

)



62. Afisarea tuturor clientilor care nu au un cont vip si participa la expozitiile angajatilor care au in inventar unelte furnizate de furnizorii care prin Suvenirurile lor produc incasari de peste 500 de lei.

select distinct c.IDClient,c.Nume,c.Prenume,c.TipClient

from Clienti c

inner join Rezervari r

on c.IDClient=r.IDClient

inner join Expozitii e

on e.IDExpozitie=r.IDExpozitie

inner join Angajati a

on a.IDAngajat=e.IDExpozitie

inner join AngajatUnealta au

on au.IDAngajat=a.IDAngajat

inner join Unelte u

on u.IDUnealta=au.IDUnealta

inner join FurnizorUnealta fu

on fu.IDUnealta=u.IDUnealta

inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

inner join FurnizorSuvenir fs

on fs.IDFurnizor=f.IDFurnizor

inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

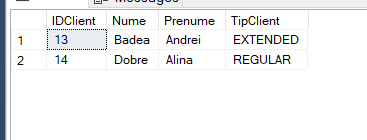
inner join IncasareSuvenir incs

on incs.IDSuvenir=s.IDSuvenir

inner join Incasari inc

on inc.IDIncasare=incs.IDIncasare

where inc.SumaIncasata>500 and c.TipClient <>'VIP'



63. Afisarea datei de inregistrare a incasarilor in care se regasesc cel putin doua suveniruri in gestiune.

select incs.DataIncasare,count(incs.IDSuvenir) as NumarSuveniruri

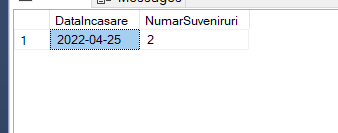
from Suveniruri s inner join IncasareSuvenir incs

on s.IDSuvenir=incs.IDSuvenir inner join Incasari i

on i.IDIncasare=incs.IDSuvenir

group by incs.DataIncasare

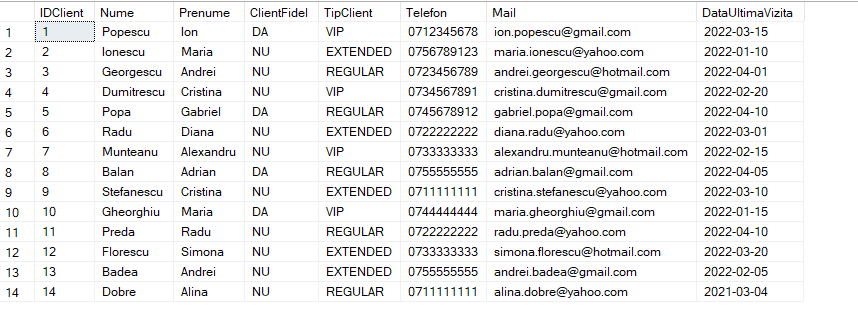
having count(incs.IDSuvenir)> 1



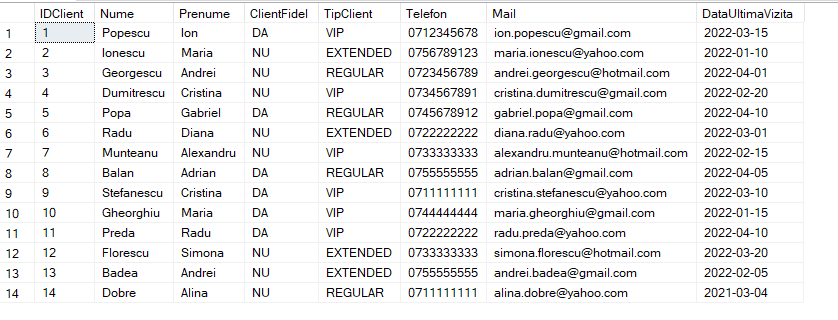
## 2.3. UPDATE

**1. Managerii decid ca acei clienti care achizitioneaza bilete la expozitia „Sala sculpturii” sa primeasca titulatura de client fidel si tipul acestuia sa devina EXTENDED.**

**Inainte:**

****

**Dupa update:**

****

**2. Luna mai a trecut asa ca data ultimei vizite pentru clienti trebuie sa fie actualizata cu cea mai recenta data a rezervarii acestora.**

declare @contor1 int;

declare @contor2 int;

set @contor1=1

set @contor2=1

while @contor2<15

begin

update Clienti

set DataUltimaVizita =

(select top(1) r.DataRezervare

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

where c.IDClient=@contor1

)

where IDClient=@contor2;

set @contor1=@contor1+1;

set @contor2=@contor2+1;

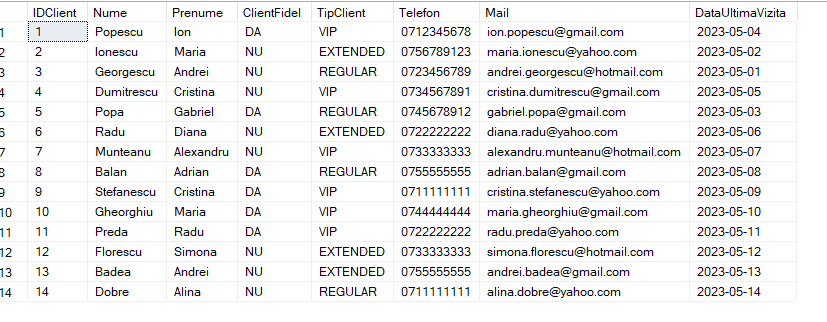
end

select\* from Clienti

**Inainte de update:**

****

**Dupa update:**

****

**3. Managerii decid o crestere salariala pentru functiile ocupate de persoanele din departamentul de ingrijire.**

update Salarii

set SalariuBrut=SalariuBrut+0.05\*SalariuBrut,SalariuNet=SalariuNet+0.05\*SalariuNet

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

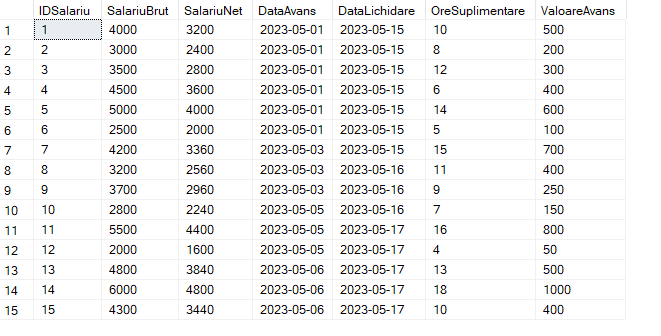
on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

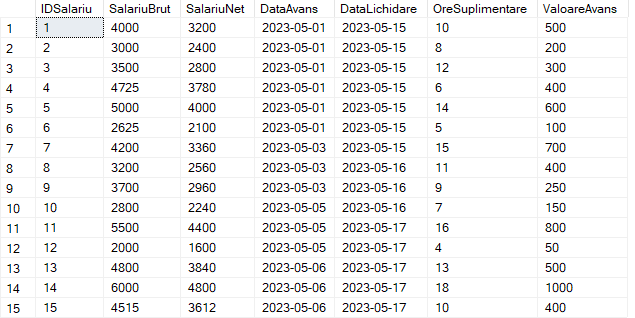
on s.IDSalariu=sf.IDSalariu

where a.Departament='Ingrijire'

**Inainte de update:**

****

**Dupa update:**

****

**4. In conformitate cu o lege, se considera ca persoanele care lucreaza in departamentul IT si in cel de Marketing sunt expusi la un risc ridicat de radiatii. Astfel se va micsora cu 2 procente impozitul pentru casa de asigurari.**

update Impozite

set ProcentAsigurareMedicala=ProcentAsigurareMedicala-2

from Angajati a inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat inner join Functii f

on fa.IDFunctie=f.IDFunctie inner join SalariuFunctie sf

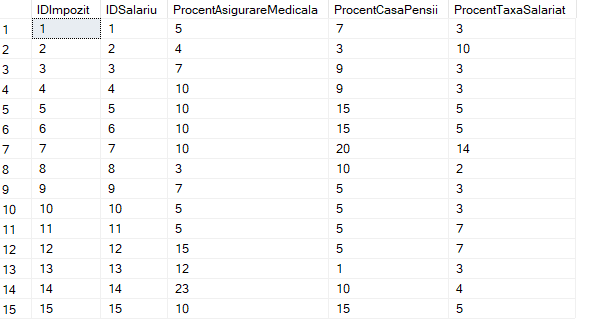
on f.IDFunctie=sf.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu inner join Impozite i

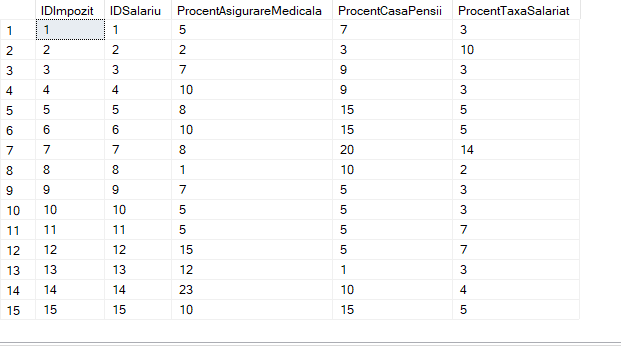
on s.IDSalariu=i.IDSalariu

where a.Departament='Marketing' or a.Departament='IT'

**Inainte de update:**

****

**Dupa update:**

****

**5. Toti furnizorii de suveniruri din Bucuresti au hotarat ca preturile obiectelor furnizate sa devina egale cu media preturilor tuturilor suvenirurilor furnizate de ei.**

update Suveniruri

set Pret=

(select sum(Pret)/count(Pret)

from Furnizori f inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

where f.Adresa like '%Bucuresti')

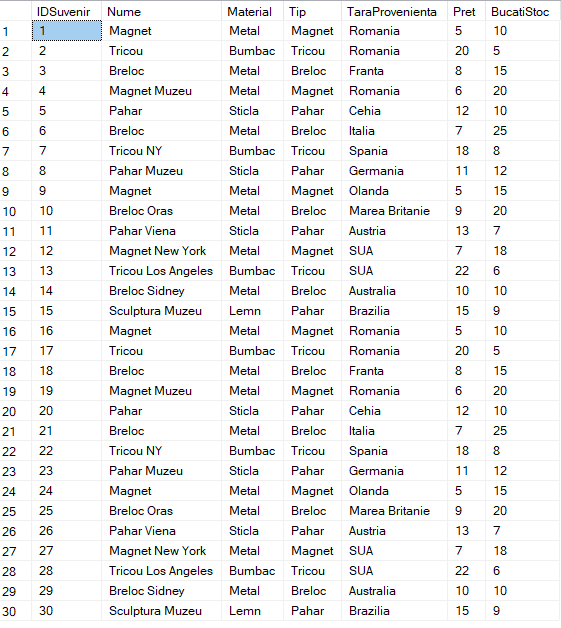
from Furnizori f inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

where f.Adresa like '%Bucuresti'

**Inainte de update:**

****

**Dupa update:**

****

**6. S-a constatat o eroare in baza de date. La cumpararea uneltelor pe care le folosesc managerii s-a optat pentru prelungirea garantiei cu 6 luni, ceea ce a fost omis in baza. Reparati eroarea.**

**Inainte de Update:**

****

**Dupa update:**

****

**7. Se observa ca data ultimelor incasari inregistrate in baza de date este foarte veche. Sa se modifice data crescator incepand de la 2 mai 2023.**

declare @contor int;

set @contor=1;

WHILE @contor<15

begin

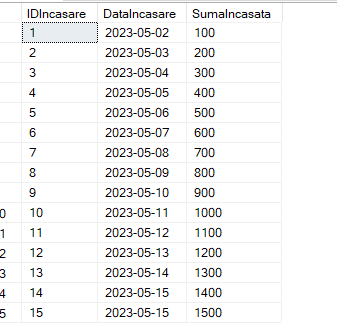
update Incasari

set DataIncasare=DATEADD(day,@contor,'2023/05/01')

where IDIncasare=@contor

set @contor=@contor+1

end

****

**8. Pentru exponatul care are in baza de date id-ul 40 sa se modifice dataAparitie pentru acesta, stiind ca a fost creat la aniversarea de 20 de ani a autorului.**

declare @dataNastereAutor DATE;

set @dataNastereAutor= (select c.DataNastere

from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on e.IDExponat=ce.IDExponat

where e.IDExponat=40)

update Exponate

set DataAparitie=DATEADD(year,20,@dataNastereAutor)

where IDExponat=40

**Inainte de update:**

****

**Dupa update:**

****

**9. Angajatii cu functia de Curatori au primit o misiune importanta. Deci acestia sunt nevoiti sa amane Expozitiile de care sunt responsabili pana la ora 15. Executati aceasta schimbare.**

update Rezervari

set OraRezervare=15

from Functii f inner join FunctieAngajat fa

on f.IDFunctie=fa.IDFunctie inner join Angajati a

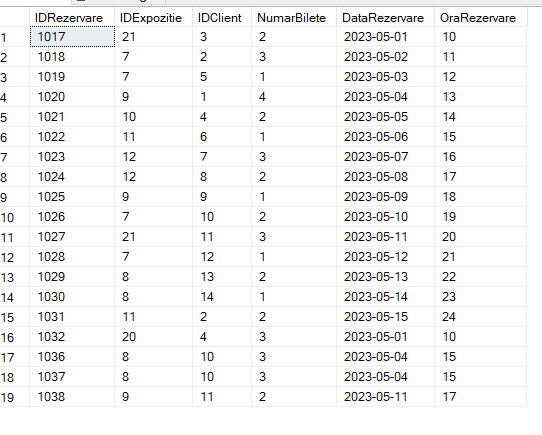
on fa.IDAngajat=a.IDAngajat inner join Expozitii e

on e.IDAngajat=a.IDAngajat inner join Rezervari r

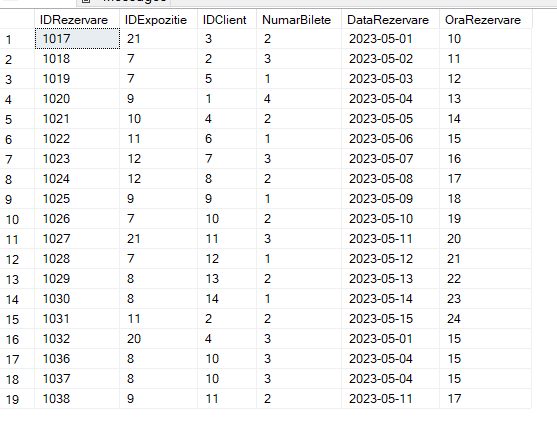
on r.IDExpozitie=e.IDExpozitie

where f.NumeFunctie='Curator'

**Inainte de update:**

****

**Dupa update:**

****

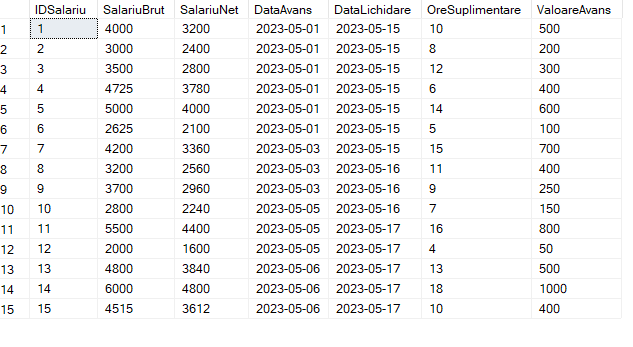
**10. Cresterea cu 10 la suta a salariilor care au un necesar de peste 10 ore suplimentare.**

update Salarii

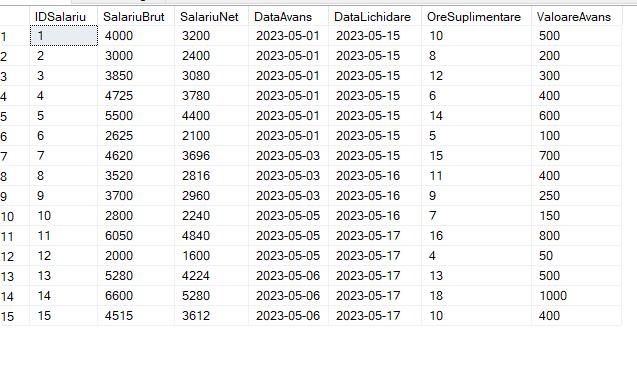
set SalariuBrut=SalariuBrut+0.1\*SalariuBrut,SalariuNet=SalariuNet+0.1\*SalariuNet

where OreSuplimentare>10

**Inainte de update:**

****

**Dupa update:**

****

**11. Diminuarea celui mai mare salariu brut cu valoarea medie a avansurilor.**

declare @AvansMediu int;

declare @celMaiMareSalariu int;

set @AvansMediu=(select avg(ValoareAvans)

from Salarii)

set @celMaiMareSalariu=(select max(SalariuBrut)

from Salarii)

update Salarii

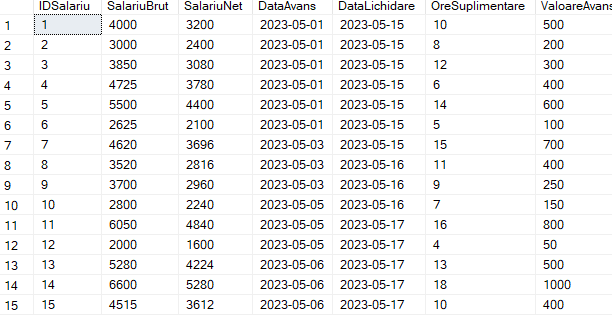
set SalariuBrut=SalariuBrut-@AvansMediu

where SalariuBrut=@celMaiMareSalariu

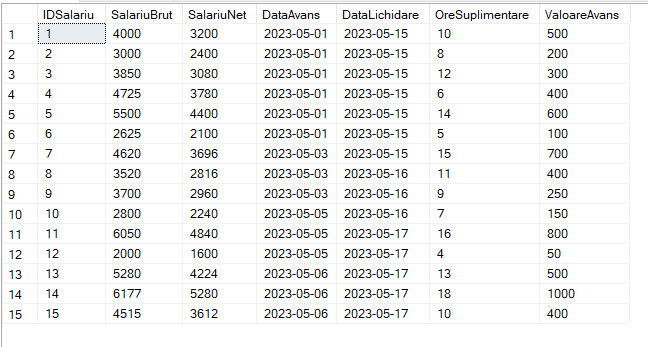
select\* from Salarii

**Inainte de update:**

**Dupa update:**

****

**Dupa update:**

****

**12. Furnizorii decid cresterea Duratei Contractului de Furnizare cu 3 ani pentru firmele acestora in cazul in care sunt mai vechi de anul 2012.**

update Furnizori

set DurataContractFurnizare=DurataContractFurnizare+3

where Year(AnActivitatePiata)<2012

**Inainte de update:**

****

**Dupa update:**

****

**13. Pentru a evita aglomeratia, angajatii care au rolul de ghid turistic vor limita numarul de locuri disponibile pentru salile de care sunt responsabili la jumatate.**

update Expozitii

set CapacitatePersoane=CapacitatePersoane/2

from Expozitii e inner join Angajati a

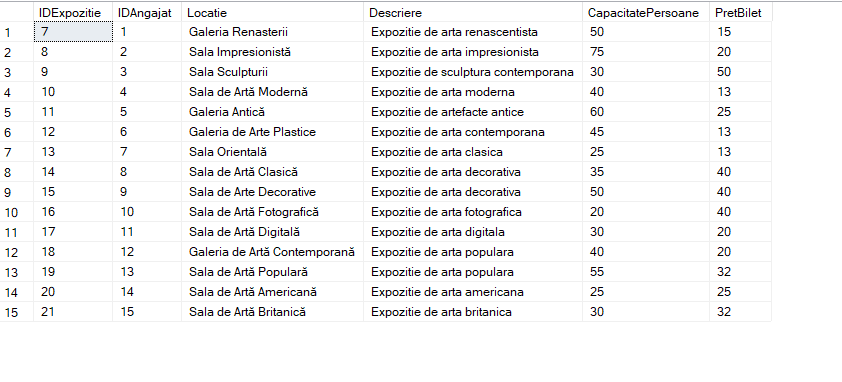
on e.IDAngajat=a.IDAngajat inner join FunctieAngajat fa

on fa.IDAngajat=a.IDAngajat inner join Functii f

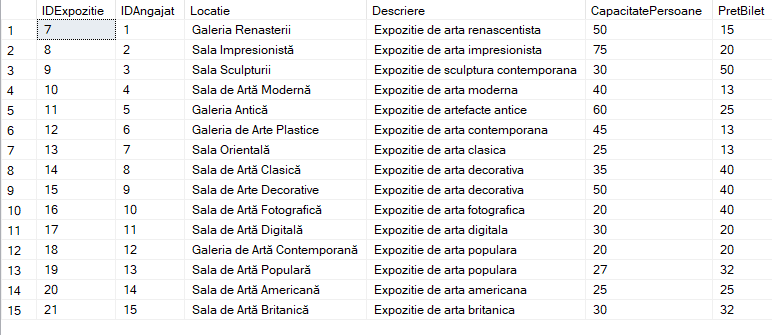
on f.IDFunctie=fa.IDFunctie

where f.NumeFunctie='Ghid turistic'

**Inainte de update:**

****

**Dupa update:**

****

**14. Cresterea salariului net cu 100 de lei pentru angajatii care au in raspundere suveniruri de tip Bijuterie.**

update Salarii

set SalariuNet=SalariuNet+100

from Salarii s inner join SalariuFunctie sf

on s.IDSalariu=sf.IDSalariu inner join Functii f

on f.IDFunctie=sf.IDFunctie inner join FunctieAngajat fa

on f.IDFunctie=fa.IDFunctie inner join Angajati a

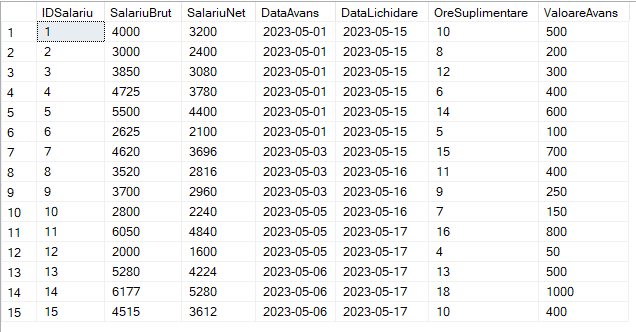
on a.IDAngajat=fa.IDAngajat inner join Expozitii e

on e.IDAngajat=a.IDAngajat inner join Exponate expo

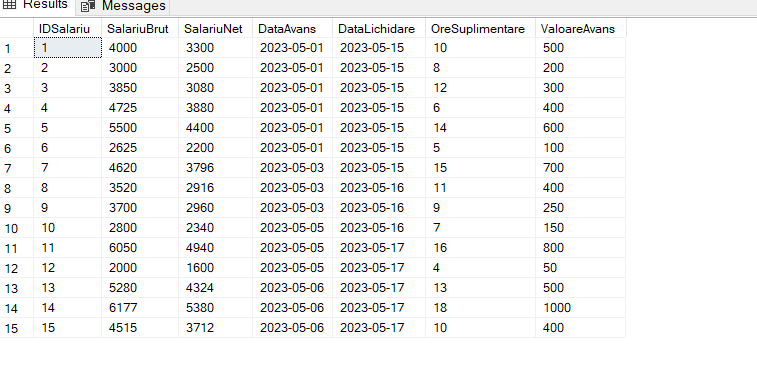
on e.IDExpozitie=expo.IDExpozitie

where expo.Tip='Bijuterie'

**Inainte de update:**

****

**Dupa update:**

**15. Adaugarea unei noi taxe in tabelul impozite pe care angajatii care au in grija exponate de tip bijuterii trebuie sa o plateasca, procentul fiind de 2%, reprezentand GarantieBijuterii.**

alter table Impozite

drop column GarantieBijuterii

alter table Impozite

add GarantieBijuterii int;

select\* from Impozite

update Impozite

set GarantieBijuterii=2

from Impozite i inner join Salarii s

on i.IDSalariu=s.IDSalariu inner join SalariuFunctie sf

on sf.IDSalariu=s.IDSalariu inner join Functii f

on f.IDFunctie=sf.IDFunctie inner join FunctieAngajat fa

on fa.IDFunctie=f.IDFunctie inner join Angajati a

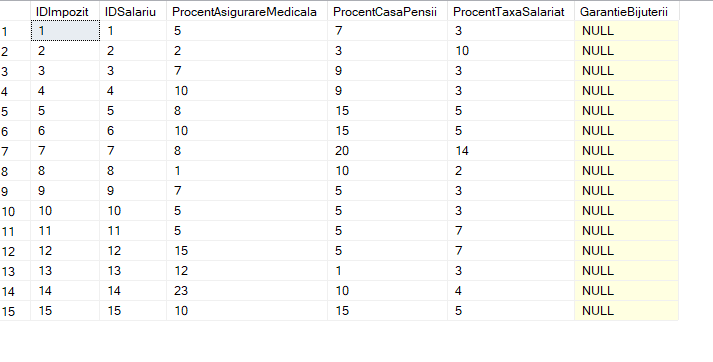
on a.IDAngajat=fa.IDAngajat inner join Expozitii e

on e.IDAngajat=a.IDAngajat inner join Exponate expo

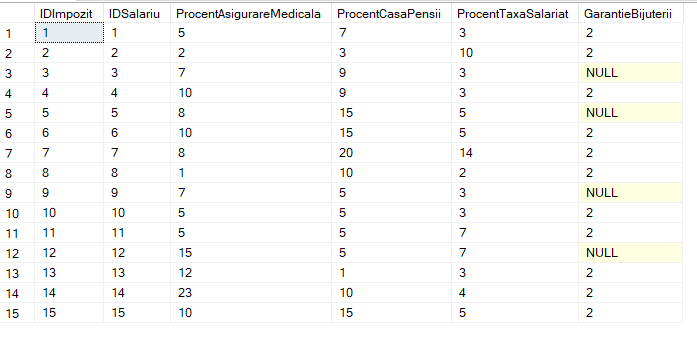
on expo.IDExpozitie=e.IDExpozitie

where expo.tip='Bijuterie'

**Inainte de udpate:**

****

**Dupa update:**

****

**16. Angajatul care are cele mai multe unelte trebuie sa gestioneze una dintre ele, ceea ce inseamna ca va fi nevoie ca ora rezervarilor clientilor pentru sala acestuia va trebui sa devina 20. Executati modificarile.**

update Rezervari

set OraRezervare=20

from Rezervari r inner join Expozitii e

on r.IDExpozitie=e.IDExpozitie inner join Angajati a

on a.IDAngajat=e.IDAngajat

where a.IDAngajat=(

select top(1) a.IDAngajat

from Angajati a inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat inner join Unelte u

on u.IDUnealta=au.IDUnealta

group by a.IDAngajat

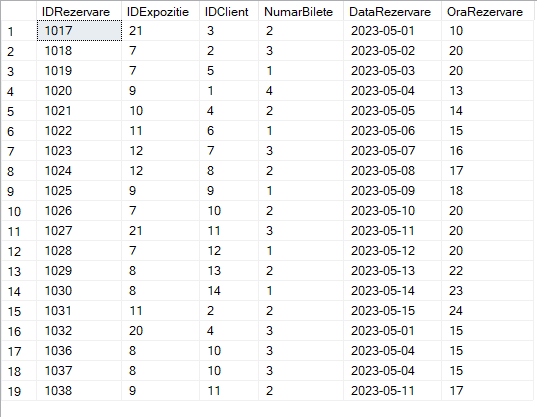
order by count(au.IDAngajat) desc

)

**Inainte de update:**

****

**Dupa update:**

****

**17. In urma cresterii inflatiei, se constata ca este necesara o crestere a preturilor biletelor cu jumatate din pretul celui mai ieftin bilet pentru expozitiile la care participa peste 5 persoane.**

declare @minim int;

set @minim=(select min(PretBilet) from Expozitii)

update Expozitii

set PretBilet=PretBilet+@minim/2

where IDExpozitie in(

select e.IDExpozitie

from Expozitii e inner join Rezervari r

on r.IDExpozitie=e.IDExpozitie

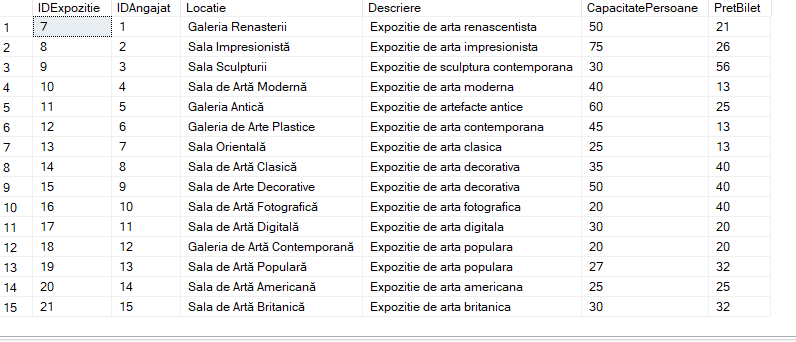
group by e.IDExpozitie

having sum(r.NumarBilete)>5

)

**Inainte de update:**

****

**Dupa update:** ****

**18. Toate Uneltele de tip recuzita de la marca Bic sau Canon vor fi catalogate drept Unelte de tip ‚Manual’.**

update Unelte

set TipUnealta='Manual'

where Marca='Canon' or Marca='Bic' and TipUnealta='Recuzita'

**Dupa update:**

****

**19. Conducerea muzeului considera ca toti clientii de tip Regular care participa la expozitiile in care sunt prezentate creatii ale artistilor specializati in Pictura, merita abonamentul de tip extended daca acestia sunt regular. Executati acest task.**

update Clienti

set TipClient='EXTENDED'

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient inner join Expozitii e

on e.IDEXpozitie=r.IDExpozitie inner join Exponate expo

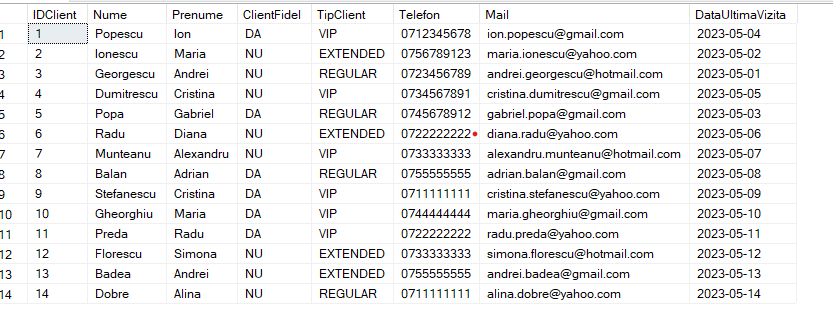
on expo.IDExpozitie=e.IDExpozitie inner join CreatorExponat ce

on ce.IDExponat=expo.IDExponat inner join Creatori cr

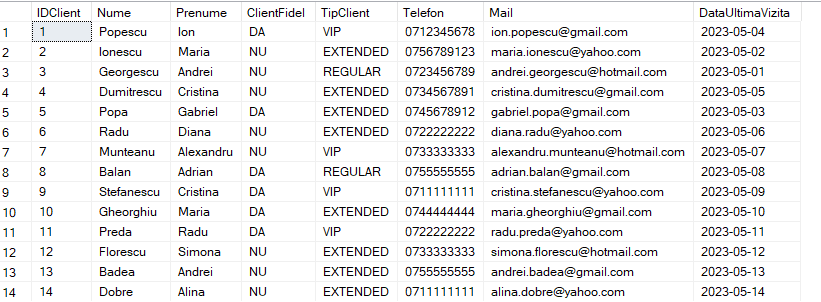
on cr.IDCreator=ce.IDCreator

where cr.TipuriPreferate='Pictura'

**Inainte:**

****

**Dupa:**

****

**20. Prelungirea contractelor angajatilor cu inca 2 ani pentru angajatii carora le expira in 2022.**

update Angajati

set DataExpirareContract=DATEADD(year,2,DataExpirareContract)

where YEAR(DataExpirareContract)=2022

**Inainte:**

****

**Dupa:**

****

## 2.4. DELETE

**1. Stergeti toti clientii care au rezervare dupa data de 5 mai la o sala la care angajat responsabil este un Curator.**

delete c

from Clienti c

inner join Rezervari r

on c.IDClient=r.IDClient

inner join Expozitii e

on e.IDExpozitie=r.IDExpozitie

inner join Angajati a

on a.IDAngajat=e.IDAngajat

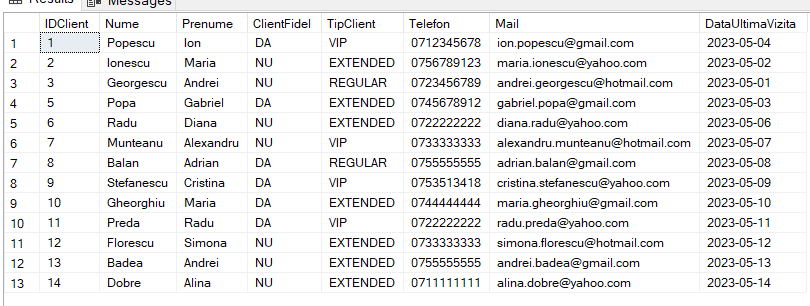
inner join FunctieAngajat fa

on fa.IDAngajat=a.IDAngajat

inner join Functii f

on f.IDFunctie=fa.IDFunctie

where f.NumeFunctie='Curator' and r.DataRezervare>'2023/05/05'

****

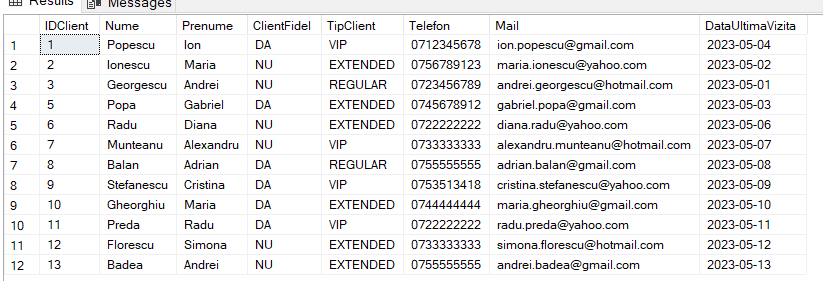
**2. Sa se stearga toti clientii care au tipul de abonament EXTENDED , numele acestora incepe cu litera ‚D’ si au contul de e-mail cu extensia @yahoo.com.**

delete

from Clienti

where Nume like 'D%' and TipClient='EXTENDED' and Mail like '%@yahoo.com'

select\* from Clienti



**3. Sa se stearga angajatul pentru care furnizorul livreaza si suvenirul „Pahar Viena”.**

delete a

from Angajati a

inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat

inner join Unelte u

on au.IDUnealta=u.IDUnealta

inner join FurnizorUnealta fu

on fu.IDUnealta=u.IDUnealta

inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

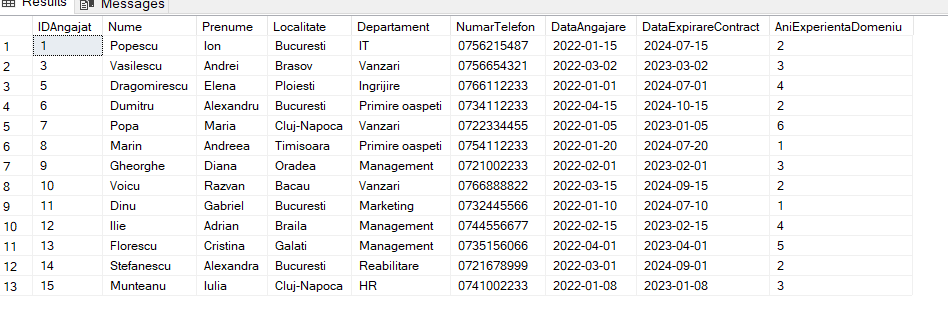
inner join FurnizorSuvenir fs

on fs.IDFurnizor=f.IDFurnizor

inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

where s.Nume='Pahar Viena'



**4. Stergerea clientului in urma caruia s-a obtinut cea mai mica incasare pe baza rezervarii acestuia.**

delete c

from Clienti c

inner join Rezervari r

on r.IDClient=c.IDClient

inner join IncasareRezervare ir

on ir.IDRezervare=r.IDRezervare

inner join Incasari inc

on ir.IDIncasare=inc.IDIncasare

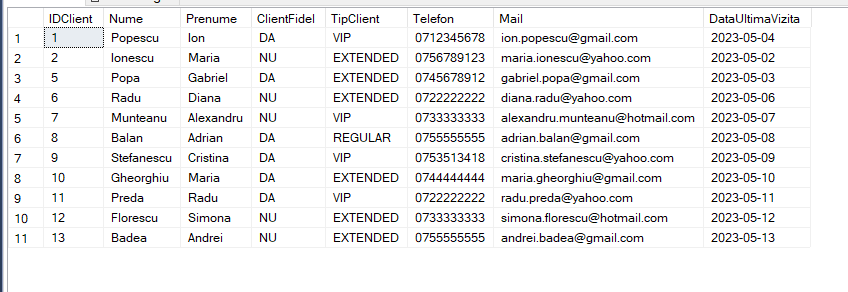
where inc.SumaIncasata=

(

select min(SumaIncasata)

from Incasari

)

****

**5. Stergerea furnizorului care este cel mai vechi pe piata de munca si nu furnizeaza nici o unealta.**

with tabelFurnizoriAn as

(

select f.IDFurnizor,f.AnActivitatePiata

from Furnizori f left join FurnizorUnealta fu

on f.IDFurnizor=fu.IDFurnizor

group by f.IDFurnizor,f.AnActivitatePiata

having count(fu.IDFurnizor)=0

)

delete f

from Furnizori f

where f.IDFurnizor=

( select top(1) t.IDFurnizor

from tabelFurnizoriAn as t

order by t.AnActivitatePiata

)

****

**6. Stergerea creatorilor ale caror exponate se afla in expozitia pentru care are rezervare clientul cu id-ul 5.**

delete c

from Creatori c

inner join CreatorExponat ce

on ce.IDCreator=c.IDCreator

inner join Exponate e

on e.IDExponat=ce.IDExponat

inner join Expozitii expo

on e.IDExpozitie=expo.IDExpozitie

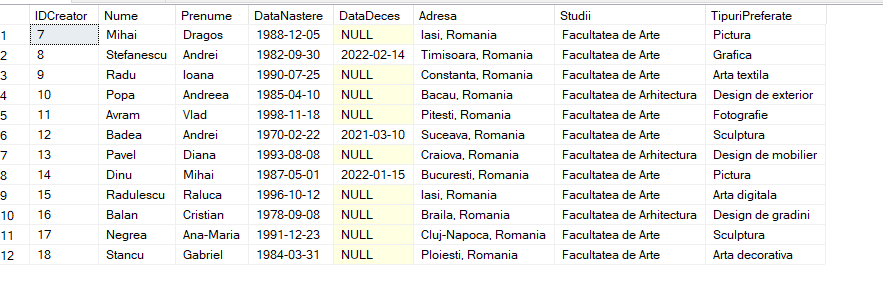
inner join Rezervari r

on r.IDExpozitie=expo.IDExpozitie

inner join Clienti cl

on cl.IDClient=r.IDClient

where cl.IDClient='5'

****

**7. Stergerea impozitelor pentru grila salariala a persoanelor care ocupa functia de Curatori.**

delete i

from Impozite i

inner join Salarii s

on s.IDSalariu=i.IDSalariu

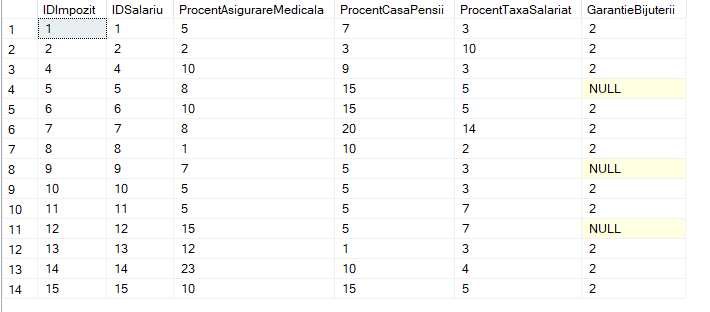
inner join SalariuFunctie sf

on sf.IDSalariu=s.IDSalariu

inner join Functii as f

on f.IDFunctie=sf.IDFunctie

where f.NumeFunctie='Curator'

****

**8. Stergerea unuia dintre exponatele creatorilor care au 5 prezente in expozitie.**

delete c

from Creatori c

where c.IDCreator=(select c.IDCreator

from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator

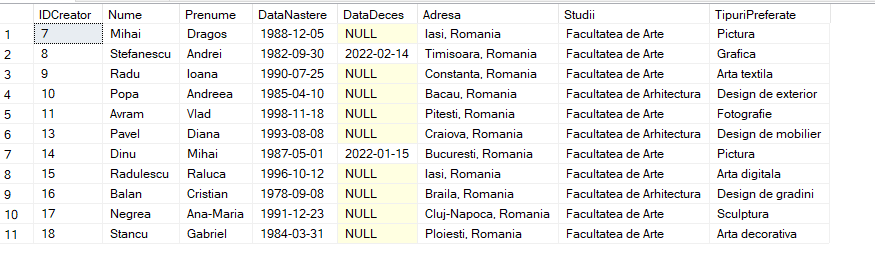
inner join Exponate e

on e.IDExponat=ce.IDExponat

group by c.IDCreator

having count(ce.IDCreator)=5

)

****

**9. Stergerea furnizorilor cu id par care furnizeaza pentru angajatii care au id impar.**

****

**10. Stergerea accesului angajatilor la Uneltele care au un id divizibil cu 5 si sunt produse de furnizori cu id-ul mai mare decat 10.**

delete au

from AngajatUnealta au

inner join Unelte u

on au.IDUnealta=u.IDUnealta

inner join FurnizorUnealta fu

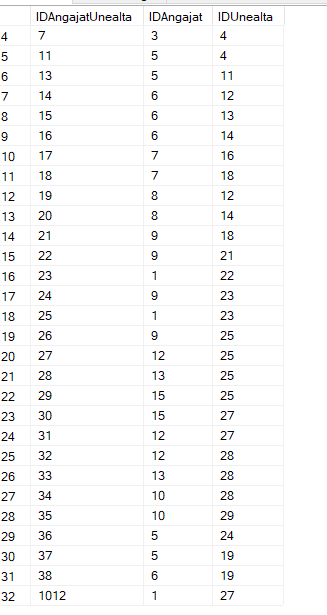
on fu.IDUnealta=u.IDUnealta

inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

where au.IDUnealta%5=0 and f.IDFurnizor>10

select \* from AngajatUnealta

****

**11. Stergerea rezervarii efectuate la sala de expozitie in care sunt prezentate exponate realizate de creatorul cu cele mai multe exponate.**

with maximum as

(

select c.IDCreator,count(ce.IDCreator) as NumarOpere

from Creatori c

inner join CreatorExponat ce

on ce.IDCreator=c.IDCreator

group by c.IDCreator

)

delete r

from Rezervari r

inner join Expozitii e

on r.IDExpozitie=e.IDExpozitie

inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

inner join CreatorExponat ce

on expo.IDExponat=ce.IDExponat

inner join Creatori c

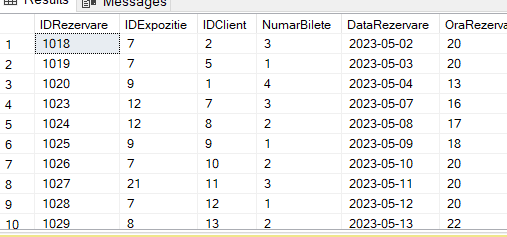
on c.IDCreator=ce.IDCreator

where c.IDCreator=( select top(1) IDCreator

from maximum

order by NumarOpere desc

)

****

**12. Stergerea salariilor pentru care functiile ocupate sunt cele ale angajatului care se afla in sala de expozitie pe care o viziteaza clientul al carui numar se termina cu cifra 4.**

delete s

from Salarii s

inner join SalariuFunctie sf

on s.IDSalariu=sf.IDSalariu

inner join Functii f

on f.IDFunctie=sf.IDFunctie

inner join FunctieAngajat fa

on f.IDFunctie=fa.IDFunctie

inner join Angajati a

on a.IDAngajat=fa.IDAngajat

inner join Expozitii e

on e.IDAngajat=a.IDAngajat

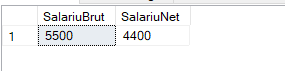
inner join Rezervari r

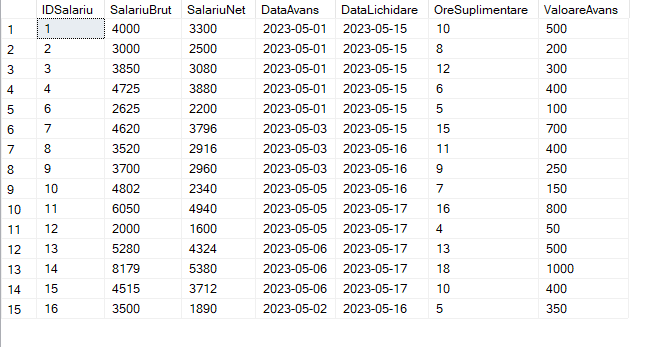
on r.IDExpozitie=e.IDExpozitie

inner join Clienti c

on c.IDClient=r.IDclient

where c.Telefon like '%4'

****

****

**13. Stergerea tuturor exponatelor care se afla in cele doua expozitii care aduc cele mai mici incasari.**

delete expo

from Exponate expo

inner join Expozitii e

on expo.IDExpozitie=e.IDExpozitie

inner join Rezervari r

on r.IDExpozitie=e.IDExpozitie

inner join IncasareRezervare ir

on ir.IDRezervare=r.IDRezervare

inner join Incasari i

on i.IDIncasare=ir.IDIncasare

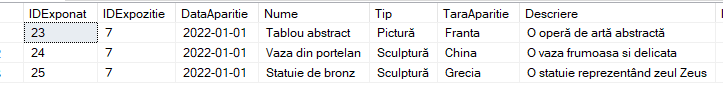
where i.SumaIncasata in (

select top(2) SumaIncasata

from Incasari

order by SumaIncasata

)

****

****

**14. Stergerea tuturor angajatilor care mai au un singur an de contract fata de data curenta(10 mai 2023) si au o singura unealta in inventar.**

delete a

from Angajati a

inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat

inner join Unelte u

on u.IDUnealta=au.IDUnealta

where a.IDAngajat in(

select a.IDAngajat

from Angajati a

inner join AngajatUnealta au

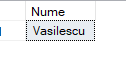
on a.IDAngajat=au.IDAngajat

group by a.IDAngajat

having count(au.IDAngajat)=1

)

and DATEDIFF(year,GETDATE(),a.DataExpirareContract) <1

****

****

**15. Stergerea tuturor uneltelor pentru care exista un singur angajat responsabil si data expirarii garantiei va fi la mai putin 6 luni de la data curenta sau au expirat deja.**

delete u

from Unelte u

where u.IDUnealta in (

select u.IDUnealta

from Unelte u

inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta

group by u.IDUnealta

having count(au.IDUnealta)=1

)

and DateDIFF(month,GETDATE(),u.DataExpirareGarantie)<6

****

## 2.5. Tranzacții

**1. Sa se insereze un nou client in baza de date, sa se realizeze un update asupra numarului acestuia de telefon si sa se afiseze campul respectiv acelui client.**

begin try

begin tran

insert into Clienti(Nume,Prenume,ClientFidel,TipClient,Telefon,Mail,DataUltimaVizita)

values('Ion','Marinel','NU','REGULAR',NULL,'ion.marinel@yahoo.com','2021/03/04')

update Clienti

set Telefon='0723436541'

where Nume='Ion' and Prenume='Marinel'

select \*

from Clienti

where Nume='Ion' and Prenume='Marinel'

commit tran

end try

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

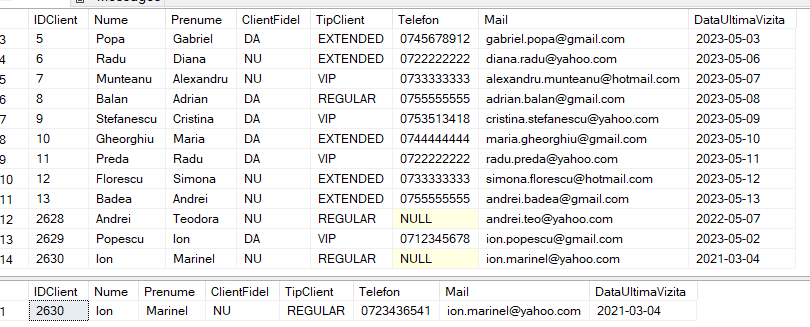
BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

****

**2. Inserarea unui angajat cu un id dat de utilizator , afisarea datelor despre acesta si stergerea angajatului cu id-ul cu 2 mai mic decat al acestuia(cel introdus urmand sa preia functiile,uneltele si sala de expozitie ale celui sters), tranzactia cuprinzand si un mecanism de tratare a exceptilor structurat.**

declare @IDAng int;

set @IDAng=10

begin try

begin tran

insert into Angajati(IDAngajat,Nume,Prenume,Localitate,Departament,NumarTelefon,DataAngajare,DataExpirareContract,AniExperientaDomeniu)

values(@IDAng,'Martin','Andrei','Bradu','IT','0732332624',GETDATE(),'2024/05/10',3)

set identity\_insert Angajati off;

update FunctieAngajat

set IDAngajat=@IDAng

where IDAngajat=@IDAng-2

update AngajatUnealta

set IDAngajat=@IDAng

where IDAngajat=@IDAng-2

delete Angajati

where IDAngajat=@IDAng-2

commit tran

end try

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

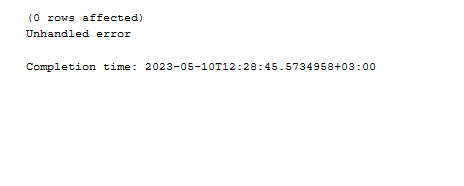
BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

****

**--Apare unhandled error deoarece nu exista in baza de date un corespondent cu id cu 2 mai mic.**

**3. Tranzactie prin care sa se introduca un suvenir, sa se actualizeze si afiseze stocul de produse de tip suvenir si sa se stearga suvenirul care produce cei mai putini bani.**

BEGIN TRY

BEGIN TRAN;

insert into Suveniruri(Nume,Material,Tip,TaraProvenienta,Pret,BucatiStoc)

values('Pantaloni negri','Bumbac','Pantaloni','China',70,20)

--

update Suveniruri

set BucatiStoc=BucatiStoc+5

where Nume <> 'Pantaloni negri'

delete

from Suveniruri

where IDSuvenir = ( select top(1) IDSuvenir

from Suveniruri

order by Pret\*BucatiStoc)

COMMIT TRAN;

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

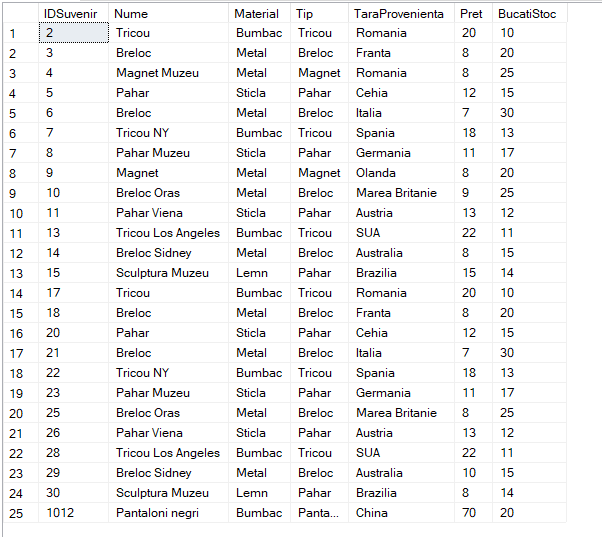
BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

****

**4. Sa se actualizeze valoarea avansului(creste cu 200 de lei) pentru angajatii din Departamentul de Management si sa se plaseze o \* in dreptul departamentului acestora, urmand afisarea.**

BEGIN TRAN

UPDATE Salarii

set ValoareAvans=ValoareAvans+200

from Salarii s inner join SalariuFunctie sf

on s.IDSalariu=sf.IDSalariu inner join Functii f

on f.IDFunctie=sf.IDFunctie inner join FunctieAngajat fa

on fa.IDFunctie=f.IDFunctie inner join Angajati a

on a.IDAngajat=fa.IDAngajat

where a.Departament='Management'

UPDATE Angajati

set Departament='Management\*'

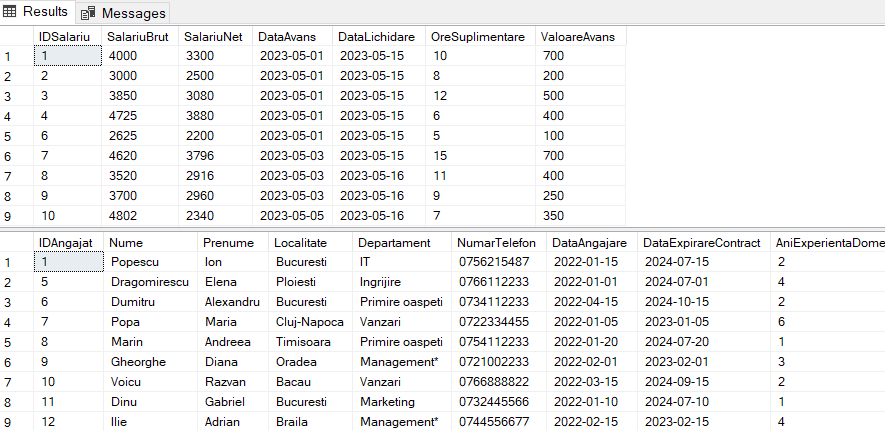
from Angajati a

where Departament='Management'

select\* from Salarii

select\* from Angajati

COMMIT TRAN

****

**5. Sa se insereze o noua rezervare realizata de clientul cu id-ul mai mic a carei sala de expozitie aleasa este ‚Sala Sculpurii’ la sala care are id-ul cu 1 mai mare decat sala sculpturii, sa se stearga rezervarea acestuia si sa se afiseze noul tabel de rezervari.**

BEGIN TRY

BEGIN TRAN

declare @IDClient int;

declare @IDExpozitie int;

set @IDExpozitie=( select IDExpozitie

from Expozitii

where Locatie='Galeria de arte Plastice')

set @IDClient=(

select top(1)c.IDClient

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

inner join Expozitii ex

on r.IDExpozitie=ex.IDExpozitie

where ex.Locatie='Galeria de arte Plastice'

order by c.IDClient

)

delete r

from Rezervari r

inner join Clienti c

on c.IDClient=r.IDClient

where c.IDClient=@IDClient

INSERT into Rezervari(IDExpozitie,IDClient,NumarBilete,DataRezervare,OraRezervare)

values(@IDExpozitie+1,@IDClient,5,'2023/05/25',13)

COMMIT TRAN

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

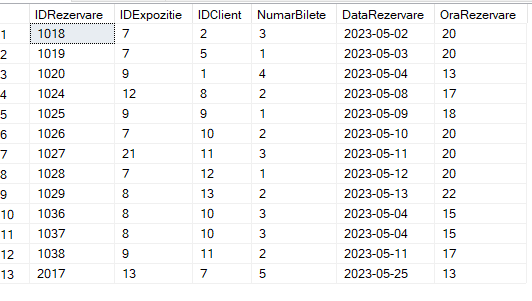
BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

****

**6. Sa se stearga accesul angajatului cu id-ul 10 la uneltele furnizate de furnizorii din Bucuresti si sa se adauge acestuia accesul la unealta realizata de furnizorul cu numarul 7.**

BEGIN TRAN

declare @errnum int;

delete AngajatUnealta

from AngajatUnealta au inner join Angajati a

on a.IDAngajat=au.IDAngajat inner join FurnizorUnealta fu

on fu.IDUnealta=au.IDUnealta inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

where au.IDAngajat=6 and f.Adresa like '%Bucuresti'

insert into AngajatUnealta(IDAngajat,IDUnealta)

values(6, (select top(1) IDUnealta

from FurnizorUnealta

where IDFurnizor=7)

)

SET @errnum = @@ERROR;

IF @errnum <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Eroare cu codul '+CAST(@errnum AS VARCHAR);

ROLLBACK

END

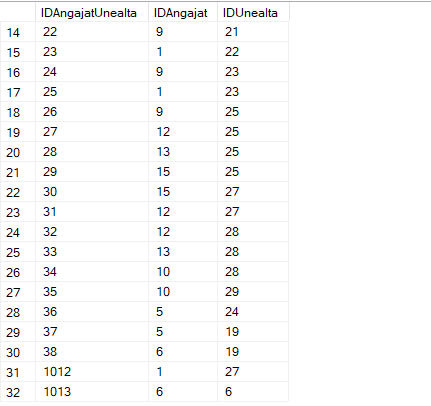
ELSE

BEGIN

select \* from AngajatUnealta

ROLLBACK

END

****

**7. Sa se insereze o noua incasare constand in preturile obtinute din toate stocurile de suveniruri disponibile la data curenta, sa se stearga cea mai veche incasare calendaristica si sa se afiseze noul tabel .**

select\* from Incasari

BEGIN TRAN

declare @errnum int;

DELETE i

from Incasari i

where i.DataIncasare = (

select top(1) DataIncasare

from Incasari

order by DataIncasare)

declare @maximum int;

set @maximum=( select max(BucatiStoc\*Pret)

from Suveniruri)

insert into Incasari(DataIncasare,SumaIncasata)

values(GETDATE(),@maximum)

SET @errnum = @@ERROR;

IF @errnum <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Eroare cu codul '+CAST(@errnum AS VARCHAR);

ROLLBACK

END

ELSE

BEGIN

select\* from Incasari

ROLLBACK

END

****

**8. Sa se realizeze o tranzactie prin care sa se introduca o noua unealta realizata de furnizorul care este cel mai vechi pe piata si sa se stearga uneltele care sunt produse de furnizorii cu contract de 12 ani.**

BEGIN TRAN

declare @errnum int;

declare @numeUnealta nvarchar(30);

declare @Marca nvarchar(30);

declare @TipUnealta nvarchar(20);

declare @DataAchizitie DATE;

declare @DataExpirareGarantie DATE;

set @numeUnealta= 'Disk Station';

set @Marca='XPenology'

set @TipUnealta='Electric'

set @DataAchizitie='2023/04/05'

set @DataExpirareGarantie='2027/10/05'

--

insert into Unelte(Nume,Marca,TipUnealta,DataAchizitie,DataExpirareGarantie)

values(@numeUnealta,@Marca,@TipUnealta,@DataAchizitie,@DataExpirareGarantie)

--

declare @idUnealta int;

set @idUnealta= (select IDUnealta

from Unelte

where Nume=@numeUnealta and Marca=@Marca and DataAchizitie=@DataAchizitie)

declare @idFurnizor int;

set @idFurnizor= (select top(1) IDFurnizor

from Furnizori

order by AnActivitatePiata)

insert into FurnizorUnealta(IDUnealta,IDFurnizor)

values(@idUnealta,@idFurnizor)

delete u

from Unelte u

inner join FurnizorUnealta fu

on u.IDUnealta=fu.IDUnealta

inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

where f.DurataContractFurnizare=12

SET @errnum = @@ERROR;

IF @errnum <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Eroare cu codul '+CAST(@errnum AS VARCHAR);

ROLLBACK

END

ELSE

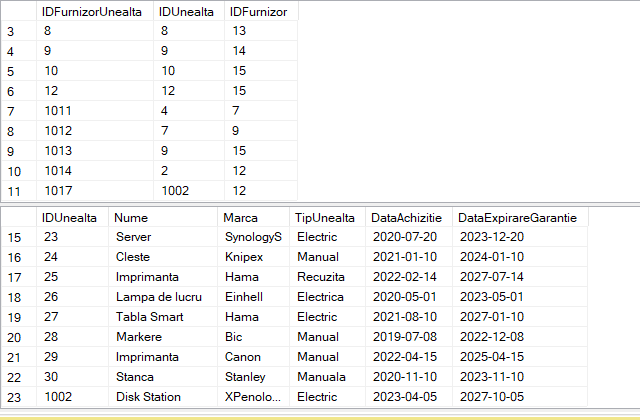
BEGIN

select\* from FurnizorUnealta

select\* from Unelte

ROLLBACK

END

****

**9. Introducerea unui nou creator care sa fie o copie a creatorului cu cel mai mare numar de opere si prenumele sau sa fie marcat cu \* la final. Acesta va primi si acces la operele vechiului creator.**

begin tran

declare @errnum int;

declare @idCreator int;

with tabelNumarExponate as

(

select c.IDCreator,count(\*) as NumarExponate

from Creatori c

inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator

group by c.IDCreator

)

select @idCreator=(select top(1) IDCreator

from tabelNumarExponate

order by NumarExponate desc

)

declare @Nume nvarchar(30)

declare @Prenume nvarchar(30)

declare @DataNastere Date

declare @DataDeces Date

declare @Adresa nvarchar(30)

declare @Studii nvarchar(30)

declare @TipuriPreferate nvarchar(30)

set @Nume= (select Nume from Creatori where IDCreator=@idCreator)

set @Prenume=(select Prenume from Creatori where IDCreator=@idCreator)

set @DataNastere=(select DataNastere from Creatori where IDCreator=@idCreator)

set @DataDeces=(select DataDeces from Creatori where IDCreator=@idCreator)

set @Adresa=(select Adresa from Creatori where IDCreator=@idCreator)

set @Studii=(select Studii from Creatori where IDCreator=@idCreator)

set @TipuriPreferate=(select TipuriPreferate from Creatori where IDCreator=@idCreator)

declare @idNou int

insert into Creatori(Nume,Prenume,DataNastere,DataDeces,Adresa,Studii,TipuriPreferate)

values(@Nume,@Prenume+'\*',@DataNastere,@DataDeces,@Adresa,@Studii,@TipuriPreferate)

set @idNou=(select IDCreator

from Creatori

where Prenume=@Prenume+'\*')

update CreatorExponat

set IDCreator=@idNou

where IDCreator=@idCreator

delete from Creatori

where IDCreator=@idCreator

SET @errnum = @@ERROR;

IF @errnum <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Eroare cu codul '+CAST(@errnum AS VARCHAR);

ROLLBACK

END

ELSE

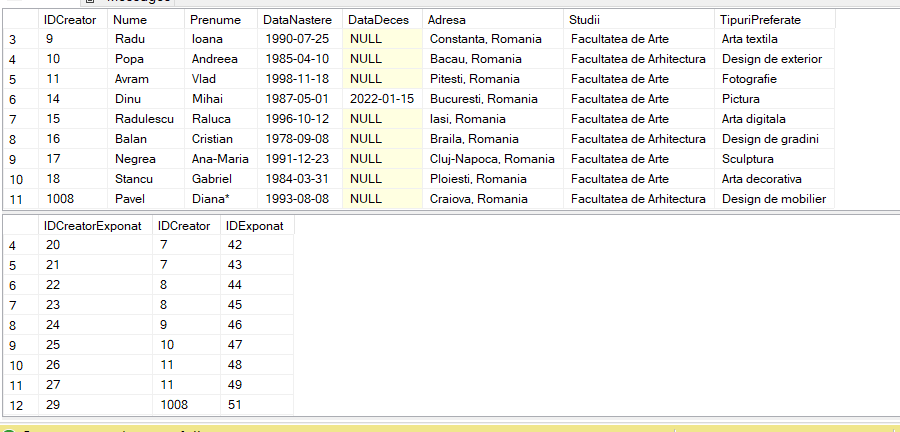
BEGIN

select\* from Creatori

select\* from CreatorExponat

ROLLBACK

END

****

**10. Actualizarea tuturor impozitelor (crestere cu 2 procente) a angajatilor care au id-ul mai mare de 10 si afisarea acestora, dar si stergerea primului impozit din lista.**

BEGIN TRY

BEGIN TRAN

update Impozite

set ProcentAsigurareMedicala=ProcentAsigurareMedicala+2,

ProcentCasaPensii=ProcentCasaPensii+2,

ProcentTaxaSalariat=ProcentTaxaSalariat+2

from Impozite i

inner join Salarii s

on s.IDSalariu=i.IDSalariu

inner join SalariuFunctie sf

on sf.IDSalariu=s.IDSalariu

inner join Functii f

on f.IDFunctie=sf.IDFunctie

inner join FunctieAngajat fa

on fa.IDFunctie=f.IDFunctie

inner join Angajati a

on a.IDAngajat=fa.IDAngajat

where a.IDAngajat>=10

delete top(1)

from Impozite

select\* from Impozite

ROLLBACK TRAN

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

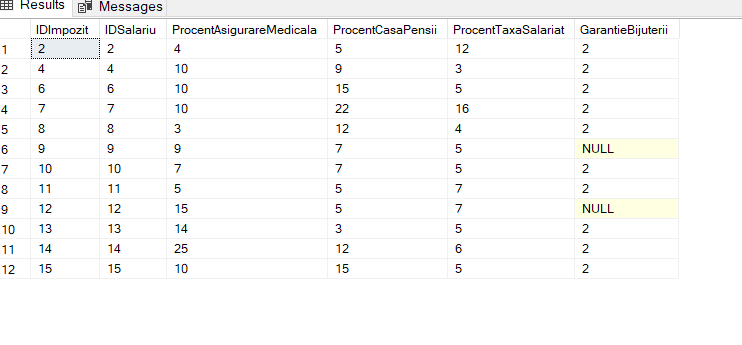
BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

****

1. Sa se insereze un nou client in baza de date si sa se faca un update asupra numarului de telefon al acestuia, in interiorul unui tranzactii.

SELECT @@TRANCOUNT;

BEGIN TRAN;

SELECT @@TRANCOUNT;

SET IDENTITY\_INSERT Clienti ON;

INSERT INTO Clienti(ClientID, Nume, Prenume, Oras)

VALUES(2015, 'Popescu', 'Mihai', 'Bucuresti');

SELECT \* FROM Clienti WHERE ClientId = 2015;

BEGIN TRAN;

SELECT @@TRANCOUNT;

UPDATE Clienti

SET Telefon = '+40766554433'

WHERE ClientId = 2015

SELECT \* FROM Clienti WHERE ClientId = 2015;

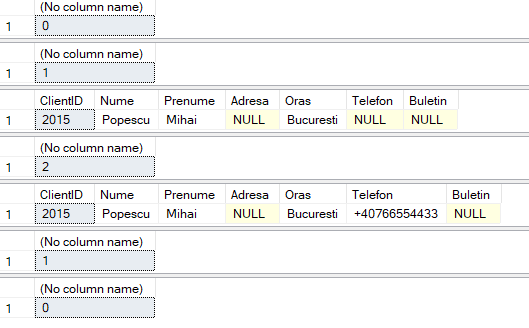
COMMIT

SET IDENTITY\_INSERT Clienti OFF;

SELECT @@TRANCOUNT;

COMMIT TRAN;

SELECT @@TRANCOUNT;



2. Sa se insereze un nou client cu id-ul 2015, sa se capteze erorile aparute la rularea de 2 ori a aceleasi tranzactii printr-o metoda nestructurata.

select \* from Clienti

DECLARE @errnum AS INT;

BEGIN TRAN;

SET IDENTITY\_INSERT Clienti ON;

INSERT INTO Clienti(ClientID, Nume, Prenume, Oras)

VALUES(2005, 'Popescu', 'Mihai', 'Bucuresti');

SET IDENTITY\_INSERT Clienti OFF;

SET @errnum = @@ERROR;

IF @errnum <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Insert into Clienti failed with error '+CAST(@errnum AS VARCHAR);

ROLLBACK

END

ELSE

BEGIN

COMMIT

END

SELECT @@TRANCOUNT

select \* from Clienti



3. Sa se rezolve exercitiul anterior cu tratarea exceptiilor printr-o metoda structurata.

BEGIN TRY

BEGIN TRAN;

SET IDENTITY\_INSERT Clienti ON;

INSERT INTO Clienti(ClientID, Nume, Prenume, Oras)

VALUES(2005, 'Popescu', 'Mihai', 'Bucuresti');

SET IDENTITY\_INSERT Clienti OFF;

COMMIT TRAN;

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;



4. Sa se reinsereze in tabelul Angajati, un angajat care care deja exista, captati eroare printr-o metoda nestructurata.

select \* from Angajati

DECLARE @errnum1 AS INT;

BEGIN TRAN;

SET IDENTITY\_INSERT Angajati ON;

INSERT INTO Angajati(AngajatID, Nume, Prenume, DataAngajarii, Oras)

VALUES(1000, 'Ionescu', 'Ioana', '2020-01-01', 'Cluj-Napoca');

SET IDENTITY\_INSERT Angajati OFF;

SET @errnum1 = @@ERROR;

IF @errnum1 <> 0 -- gestioneaaa eroarea

BEGIN

PRINT 'Insert into Angajati failed with error '+CAST(@errnum1 AS VARCHAR);

ROLLBACK

END

ELSE

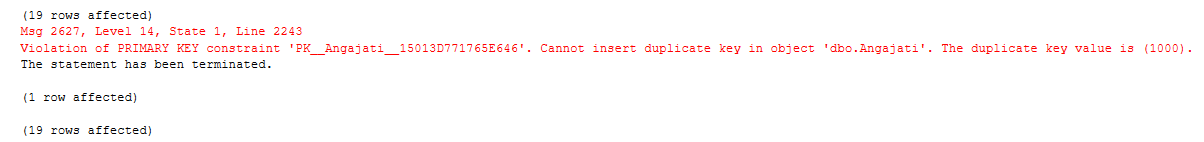
BEGIN

COMMIT

END

SELECT @@TRANCOUNT

select \* from Angajati



5. Sa se rezolve exercitiul anterior folosind o metoda structurata de tratare a erorilor.

BEGIN TRY

BEGIN TRAN;

SET IDENTITY\_INSERT Angajati ON;

INSERT INTO Angajati(AngajatID, Nume, Prenume, DataAngajarii, Oras)

VALUES(1000, 'Ionescu', 'Ioana', '2020-01-01', 'Cluj-Napoca');

SET IDENTITY\_INSERT Angajati OFF;

COMMIT TRAN;

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

BEGIN

PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;



6. Dupa mai multe discutii, in meniul restaurantului se introduc mai multe feluri de mancare. Inserati „Supa crema de morcovi”, o data cu id 300 si o data cu id 307, observati ce rezulta in urma rularii comenzii. Realizati aceste modificari si captati posibilele erori aparute.

select \* from Restaurant

BEGIN TRY

BEGIN TRAN;

SET IDENTITY\_INSERT Restaurant ON;

INSERT INTO Restaurant(MeniuID, DetaliiMeniu)

VALUES(310, 'Supa crema de morcovi');

INSERT INTO Restaurant(MeniuID, DetaliiMeniu)

VALUES(300, 'Supa crema de morcovi');

SET IDENTITY\_INSERT Restaurant OFF;

COMMIT TRAN;

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

BEGIN

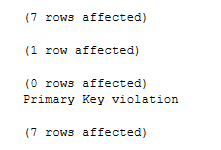
PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

select \* from Restaurant



7. Sa se efectueze inserarea a doi angajati noi, unul avand id-ul 1000, iar celalalt 1030. Observati ce se intampla in urma rularii si captati posibilele erori printr-o metoda nestructurata.

DECLARE @errnum2 AS int;

BEGIN TRAN;

SET IDENTITY\_INSERT Angajati ON;

-- Insert #1 va esua din cauza cheii primare duplicate

INSERT INTO Angajati(AngajatID, Nume, Prenume, DataAngajarii, Oras)

VALUES(1000, 'Vasile', 'Ioana', '2020-01-01', 'Cluj-Napoca');

SET @errnum2 = @@ERROR;

IF @errnum2 <> 0

BEGIN

IF @@TRANCOUNT > 0 ROLLBACK TRAN;

PRINT 'Insert #1 into Products failed: error '+ CAST(@errnum2 AS VARCHAR);

END;

-- Insert #2 se va executa cu succes

INSERT INTO Angajati(AngajatID, Nume, Prenume, DataAngajarii, Oras)

VALUES(1030, 'Marinescu', 'Maria', '2020-01-01', 'Cluj-Napoca');

SET @errnum2 = @@ERROR;

IF @errnum2 <> 0

BEGIN

IF @@TRANCOUNT > 0 ROLLBACK TRAN;

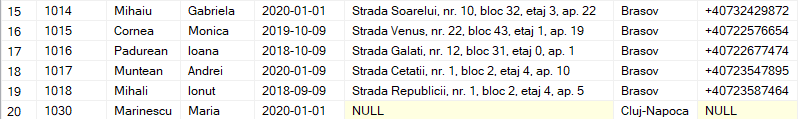
PRINT 'Insert #2 into Products failed: error '+ CAST(@errnum2 AS VARCHAR);

END;

SET IDENTITY\_INSERT Angajati OFF;

IF @@TRANCOUNT > 0 COMMIT TRAN;





8. Se incearca adaugarea unei noi activitati de agrement, „volei”. Scrieti tranzactia aferenta acestei modificari si captati eventualele erori aparute.

select \* from Agrement

BEGIN TRY

BEGIN TRAN;

SET IDENTITY\_INSERT Agrement ON;

INSERT INTO Agrement(AgrementID, NumeActivitate)

VALUES(407, 'Volei');

SET IDENTITY\_INSERT Agrement OFF;

COMMIT TRAN;

END TRY

BEGIN CATCH

IF ERROR\_NUMBER() = 2627 -- Duplicate key violation

BEGIN

PRINT 'Primary Key violation';

END

ELSE IF ERROR\_NUMBER() = 547 -- Constraint violations

BEGIN

PRINT 'Constraint violation';

END

ELSE

BEGIN

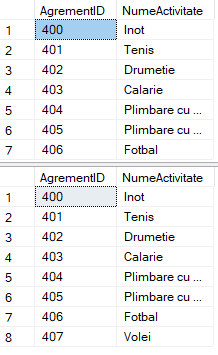
PRINT 'Unhandled error';

END;

IF @@TRANCOUNT > 0 ROLLBACK TRANSACTION;

END CATCH;

select \* from Agrement



9. Prin intermediul unei tranzactii sa se modifice pretul tuturor sericiilor care incep cu litera „D” sau „S”. Pretul va creste cu 5%, iar in dreptul numelui se va afisa „\*”.

BEGIN TRAN;

SELECT \* FROM Preturi;

UPDATE Preturi

SET Valoare = Valoare + Valoare\*0.05

WHERE NumeServiciu LIKE '[DS]%'

SELECT \* FROM Preturi;

BEGIN TRAN;

UPDATE Preturi

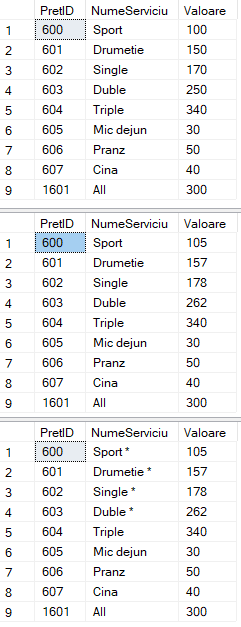
SET NumeServiciu = NumeServiciu + ' \*'

WHERE NumeServiciu LIKE '[DS]%'

SELECT \* FROM Preturi;

COMMIT

COMMIT TRAN;



10. Sa se modifice valoare tuturor rezervarilor, efectuate in prima jumatate a anului, discountul va creste cu 10%, altfel va scadea cu 1%. Scrieti tranzactia necesara acestei operatiuni.

BEGIN TRAN

SELECT \* FROM RezervariClienti;

UPDATE RezervariClienti

SET Discount = Discount + 0.1

WHERE MONTH(DataEfectuarii) > 6

SELECT \* FROM RezervariClienti;

BEGIN TRAN;

UPDATE RezervariClienti

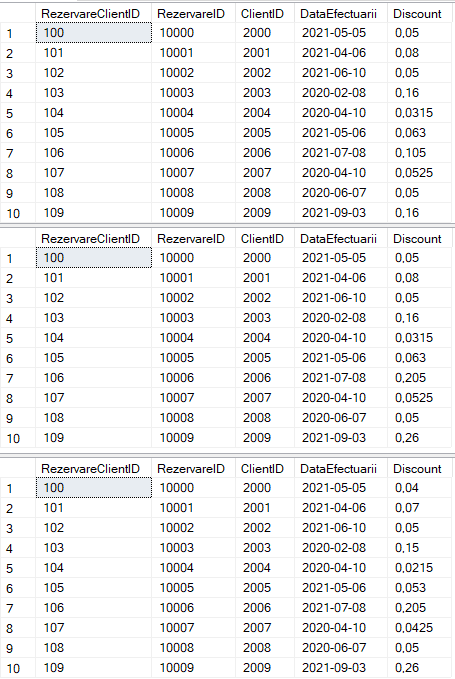
SET Discount = Discount - 0.01

WHERE MONTH(DataEfectuarii) < 6

SELECT \* FROM RezervariClienti;

COMMIT

COMMIT TRAN;



## 2.6. CTE-uri

**1. Afisarea numelui ,prenumelui si id-ului angajatului care plateste cei mai multi bani pentru casa de pensii.**

with maxim as(

select max(s.SalariuBrut\*i.ProcentCasaPensii/100) as Maxim

from Salarii s

inner join Impozite i

on s.IDSalariu=i.IDSalariu

)

select a.Nume,a.Prenume,m.Maxim

from Angajati a

inner join FunctieAngajat fa

on fa.IDAngajat=a.IDAngajat

inner join Functii f

on f.IDFunctie=fa.IDFunctie

inner join SalariuFunctie sf

on sf.IDFunctie=f.IDFunctie

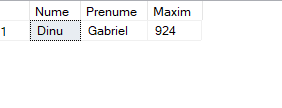
inner join Salarii s

on s.IDSalariu=sf.IDSalariu

inner join Impozite i

on i.IDSalariu=s.IDSalariu

inner join maxim m on m.Maxim=s.SalariuBrut\*i.ProcentCasaPensii/100

****

**2. Statul roman vrea ca doar procentele prezente in baza de date(AsigurareMedicala,CasaPensii...) sa ramana in vigoare din toate impozitele doar pentru persoanele care ocupa functia de manager sau asistent manager. Calculati care ar fi salariul in cazul in care doar acestea vor reprezenta „reducerea” salariului brut( se va afisa si numele angajatilor care au aceste functii).**

with ImpoziteTotale as

(

select i.IDSalariu,ProcentAsigurareMedicala+ProcentTaxaSalariat+ProcentCasaPensii as ImpozitFinal

from Impozite i

inner join Salarii s

on s.IDSalariu=i.IDSalariu

inner join SalariuFunctie sf

on sf.IDSalariu=s.IDSalariu

inner join Functii f

on sf.IDFunctie=f.IDFunctie

where f.NumeFunctie='Asistent Manager' or f.NumeFunctie='Manager'

)

select a.Nume,a.Prenume,s.SalariuBrut-s.SalariuBrut\*it.ImpozitFinal/100 as NoulSalariu

from Angajati a

inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat

inner join Functii f

on f.IDFunctie=fa.IDFunctie

inner join SalariuFunctie sf

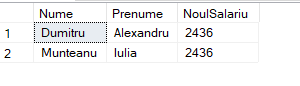
on sf.IDFunctie=f.IDFunctie

inner join Salarii s

on s.IDSalariu=sf.IDFunctie

inner join ImpoziteTotale it

on it.IDSalariu=s.IDSalariu

****

**3. Afisarea tuturor furnizorilor care produc suveniruri in urma carora se obtin incasarii de peste 500 de lei si ziua incasarii este divizibila cu 3.**

with MaximIncasari as

(

select IDIncasare

from Incasari

where SumaIncasata>500 and Day(DataIncasare)%3=0

)

select f.NumeFurnizor,f.CodSRL,f.UltimaLivrare

from Furnizori f

inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor

inner join Suveniruri s

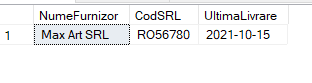
on s.IDSuvenir=fs.IDSuvenir

inner join IncasareSuvenir incs

on incs.IDSuvenir=s.IDSuvenir

inner join MaximIncasari mi

on mi.IDIncasare=incs.IDIncasare

****

**4. Afisarea tuturor uneltelor care au un numar maxim de angajati cu dreptul sa le utilizeze.**

with tabel as(

select u.IDUnealta,count(\*) as NumarAngajatiAcces

from Unelte u

inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta

inner join Angajati a

on a.IDAngajat=au.IDAngajat

group by u.IDUnealta

)

select u.IDUnealta,u.Nume

from Unelte u

inner join AngajatUnealta au

on u.IDUnealta=au.IDUnealta

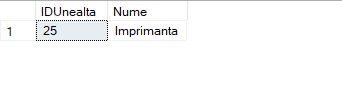
inner join Angajati a

on a.IDAngajat=au.IDAngajat

group by u.IDUnealta,u.Nume

having count(\*)=( select max(NumarAngajatiAcces)

from tabel)

****

**5. Sa se afiseze toti furnizorii care produc un numar minim de suveniruri pentru care se obtin incasari de sub 500 de lei.**

with tabelSuveniruriMinime as

(

select f.IDFurnizor,count(\*) as NumarSuveniruri

from Furnizori f

inner join FurnizorSuvenir fs

on fs.IDFurnizor=f.IDFurnizor

inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

inner join IncasareSuvenir incs

on s.IDSuvenir=incs.IDSuvenir

inner join Incasari inc

on inc.IDIncasare=incs.IDIncasare

where inc.SumaIncasata<500

group by f.IDFurnizor

)

select f.NumeFurnizor

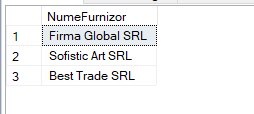
from Furnizori f

inner join tabelSuveniruriMinime tsm

on tsm.IDFurnizor=f.IDFurnizor

where tsm.NumarSuveniruri= (select min(NumarSuveniruri)

from tabelSuveniruriMinime)

****

# 3. Obiecte ale bazei de date

## 3.1. VIEW-uri

1. Sa se creeze un view care sa contina numele angajatilor,prenumele angajatilor,Numarul de telefon al acestora,departamentul de care apartin, salariul lunar total obtinut de acestia, cat si numarul de functii ocupate in cadrul muzeului.

if object\_ID('SalariereAngajati','V') is not null

drop view SalariereAngajati

go

create view SalariereAngajati

as

select a.Nume,a.Prenume,max(a.NumarTelefon) as NumarTelefon,max(a.Departament) as Departament,

sum(s.SalariuNet) as SalariuTotal,count(fa.IDAngajat) as NumarFunctii

from Angajati a

inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat

inner join Functii f

on f.IDfunctie=fa.IDFunctie

inner join SalariuFunctie sf

on sf.IDFunctie=f.IDFunctie

inner join Salarii s

on s.IDSalariu=sf.IDSalariu

group by a.Nume,a.Prenume



2. Sa se creeze un view care sa contina date referitoare la angajati si sa aiba in plus numarul de bijuterii pe care acestia le au in cadrul expozitiilor la care participa ca organizatori.

if object\_ID('AngajatiBijuterii','V') is not null

drop view AngajatiBijuterii

go

create view AngajatiBijuterii

as

select a.IDAngajat,max(a.Nume) as Nume,max(a.Prenume) as Prenume,max(a.NumarTelefon) as NumarTelefon

,max(a.Localitate) as Localitate,count(\*) as NumarBijuterii

from Angajati a

inner join Expozitii e

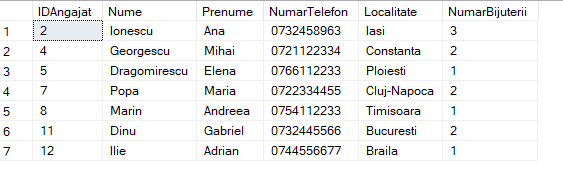
on a.IDAngajat=e.IDAngajat

inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

where expo.Tip='Bijuterie'

group by a.IDAngajat



3. Sa se creeze un view in care sa se afiseze clientii care participa la expozitiile din 2 locatii diferite( cu precizarea locatiei):Galeria Renasterii sau Sala Sculpturii si numarul de exponate prezente in acea sala.

if object\_ID('ClientiExponate','V') is not null

drop view ClientiExponate

go

create view ClientiExponate

as

select c.IDClient,max(c.Nume) as Nume,max(c.Prenume) as Prenume ,max(e.Locatie) as Locatie,count(e.IDExpozitie) as NumarExponate

from Clienti c

inner join Rezervari r

on r.IDClient=c.IDClient

inner join Expozitii e

on e.IDExpozitie=r.IDExpozitie

inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

where e.Locatie='Sala Sculpturii'

group by c.IDClient

union

select c.IDClient,max(c.Nume) as Nume,max(c.Prenume) as Prenume ,max(e.Locatie) as Locatie,count(e.IDExpozitie) as NumarExponate

from Clienti c

inner join Rezervari r

on r.IDClient=c.IDClient

inner join Expozitii e

on e.IDExpozitie=r.IDExpozitie

inner join Exponate expo

on expo.IDExpozitie=e.IDExpozitie

where e.Locatie='Galeria Renasterii'

group by c.IDClient



4. Sa se creeze un view care sa contina toate suvenirurile si furnizorii acestora, pentru care s-au inregistrat incasari de peste 500 de lei, incasarile produse de pe urma vanzarilor acestora , cat si sumele care mai pot fi obtinute pe acestea.

if object\_ID('IncasariSuveniruri','V') is not null

drop view IncasariSuveniruri

go

create view IncasariSuveniruri

as

select f.NumeFurnizor,s.Nume,s.Tip,s.Pret\*s.BucatiStoc as ValoareStocRamas,i.SumaIncasata,

s.Pret\*s.BucatiStoc+i.SumaIncasata as SumaPosibilaIncasare

from Furnizori f

inner join FurnizorSuvenir fs

on fs.IDFurnizor=f.IDFurnizor

inner join Suveniruri s

on fs.IDSuvenir=s.IDSuvenir

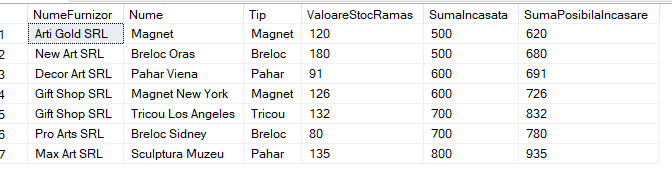
inner join IncasareSuvenir incs

on s.IDSuvenir=incs.IDSuvenir

inner join Incasari i

on i.IDIncasare=incs.IDIncasare

where i.SumaIncasata>=500



5. Sa se creeze un view care sa surprinda creatorii exponatelor din muzeu ,operele acestora si angajatii care raspund de operele lor prezente in expozitie.

if object\_ID('AngajatiResponsabili','V') is not null

drop view AngajatiResponsabili

go

create view AngajatiResponsabili

as

select c.IDCreator,c.Nume,c.Prenume,c.TipuriPreferate,e.Nume as Exponat,e.Descriere,a.Nume as NumeAngajat,a.Prenume as PRenumeAngajat,a.NumarTelefon

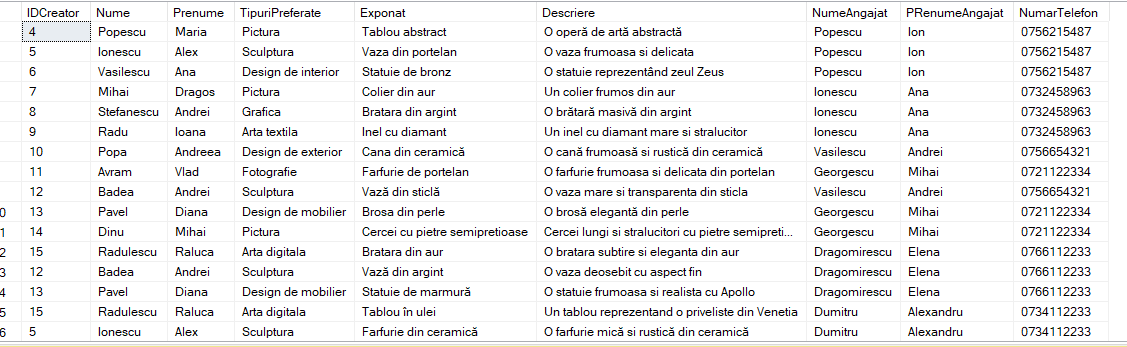
from Creatori c inner join CreatorExponat ce

on c.IDCreator=ce.IDCreator inner join Exponate e

on ce.IDExponat=e.IDExponat inner join Expozitii expo

on expo.IDExpozitie=e.IDExpozitie inner join Angajati a

on a.IDAngajat=expo.IDAngajat



6. Sa se creeze un view care sa contina numele angajatului,prenumele acestuia,numarul de telefon, cat si numarul de unelte in posesia carora acesta intra.

if object\_ID('AngajatiUnelte','V') is not null

drop view AngajatiUnelte

go

create view AngajatiUnelte

as

select max(a.Nume) as Nume,max(a.Prenume) as Prenume,max(a.NumarTelefon) as NumarTelefon,count(au.IDAngajat) as UnelteInventar

from Angajati a inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat inner join Unelte u

on u.IDUnealta=au.IDUnealta

group by a.IDAngajat



7. Sa se creeze un view care sa contina numele Functiei,Salariul Brut,Salariul Net si procentul total al impozitului specific acelei functii pentru functiile de executie.

if object\_ID('FunctiiImpozit','V') is not null

drop view FunctiiImpozit

go

create view FunctiiImpozit

as

select f.NumeFunctie,s.SalariuBrut,s.SalariuNet,i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat

as ImpozitTotal,

s.SalariuBrut-s.SalariuBrut\*(i.ProcentAsigurareMedicala+i.ProcentCasaPensii+i.ProcentTaxaSalariat)/100 as Salariu

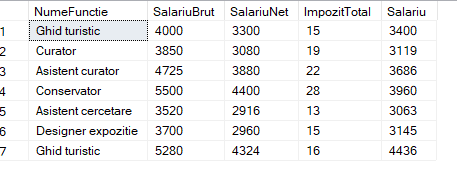
from Functii f inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie inner join Salarii s

on s.IDSalariu=sf.IDSalariu inner join Impozite i

on i.IDSalariu=s.IDSalariu

where f.Executie='DA'



## 3.2. Proceduri stocate

**1. Sa se afiseze exponatele si detaliile acestora pe care a avut ocazia sa le vada un anumit client recunoscut dupa nume si prenume.**

IF OBJECT\_ID('ClientExponate', 'P') is not null

DROP PROC ClientExponate

GO

CREATE PROC ClientExponate

@nume as nvarchar(40),

@prenume as nvarchar(40)

AS

BEGIN

select expo.Nume,expo.Descriere

from Clienti c

inner join Rezervari r

on c.IDClient=r.IDClient

inner join Expozitii e

on r.IDExpozitie=e.IDExpozitie

inner join Exponate expo

on e.IDExpozitie=expo.IDExpozitie

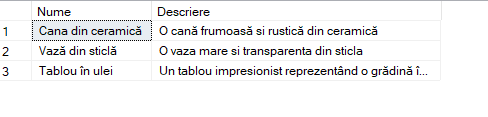
where c.Nume=@nume and c.Prenume=@prenume

END

exec ClientExponate

@nume='Preda',

@prenume='Radu'



**2. Sa se afiseze angajatii care au in evidenta un numar exact de n unelte si furnizorii acestor unelte.**

if object\_id('ProceduraAngajatiUnelte','P') is not null

drop procedure ProceduraAngajatiUnelte

go

create procedure ProceduraAngajatiUnelte

@numarUnelte int

as

begin

select a.IDAngajat,a.Nume,a.Prenume,a.NumarTelefon

from Angajati a

inner join AngajatUnealta au

on a.IDAngajat=au.IDAngajat

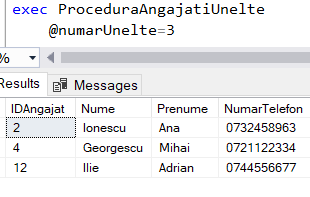
inner join Unelte u

on au.IDUnealta=u.IDUnealta

group by a.IDAngajat,a.Nume,a.Prenume,a.NumarTelefon

having count(au.IDAngajat)=@numarUnelte

end

****

**3. Sa se afiseze furnizorii care livreaza un numar exact de n suveniruri si un numar exact de m unelte pentru angajati.**

if object\_id('ProceduraFurnizoriNM','P') is not null

drop procedure ProceduraFurnizoriNM

go

create procedure ProceduraFurnizoriNM

@numarUnelte int,

@numarSuveniruri int

as

begin

select f.NumeFurnizor

from Furnizori f

inner join FurnizorSuvenir fs

on f.IDFurnizor=fs.IDFurnizor

inner join Suveniruri s

on s.IDSuvenir=fs.IDSuvenir

inner join FurnizorUnealta fu

on fu.IDFurnizor=f.IDFurnizor

inner join Unelte u

on fu.IDUnealta=u.IDUnealta

where f.IDFurnizor in

(

select f.IDFurnizor

from Suveniruri s inner join FurnizorSuvenir fs

on fs.IDSuvenir= s.IDSuvenir

inner join Furnizori f

on f.IDFurnizor=fs.IDFurnizor

group by f.IDFurnizor

having count(fs.IDFurnizor)=@numarSuveniruri

)

and u.IDUnealta in

(

select f.IDFurnizor

from Unelte u inner join FurnizorUnealta fu

on u.IDUnealta=fu.IDUnealta

inner join Furnizori f

on f.IDFurnizor=fu.IDFurnizor

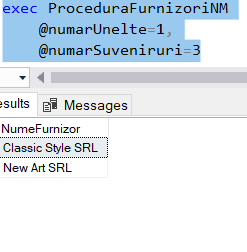
group by f.IDFurnizor

having count(fu.IDFurnizor)=@numarUnelte

)

group by f.NumeFurnizor

end

****

**4. Afisarea tuturor clientilor cu id-ul par care au rezervat un numar de peste n bilete intr o sala cu o capacitate de peste m locuri si un bilet sub k lei si a locatiei in care se afla, cat si a descrierii locatiei.**

if object\_id('ProceduraClientiSaliRezervari','P') is not null

drop procedure ProceduraClientiSaliRezervari

go

create procedure ProceduraClientiSaliRezervari

@numarBileteMinim int,

@capacitateSalaMinima int,

@pretMaximBilet int

as

begin

select c.IDClient,c.Nume,c.Prenume,e.Locatie

from Clienti c inner join Rezervari r

on c.IDClient=r.IDClient

inner join Expozitii e

on e.IDExpozitie=r.IDExpozitie

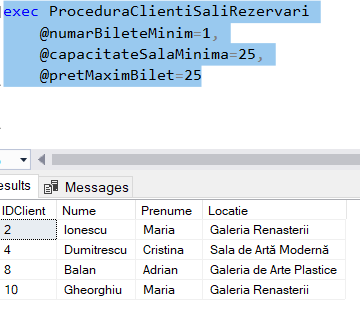
where r.IDClient%2=0 and

r.NumarBilete>@numarBileteMinim and

e.CapacitatePersoane>@capacitateSalaMinima and

e.PretBilet<@pretMaximBilet

end

****

**5. Afisarea numarului de exponate grupate pe data de aparitie care pot fi vazute de catre clientii care au o rezervare intre doua date oferite de la tastatura.**

if object\_id('ProceduraExponateData','P') is not null

drop procedure ProceduraExponateData

go

create procedure ProceduraExponateData

@DataInitiala Date,

@DataFinala Date

as

begin

select expo.DataAparitie,count(\*) as NumarExponate

from Expozitii e

inner join Exponate expo

on e.IDExpozitie=expo.IDExpozitie

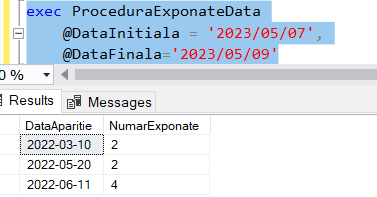
inner join Rezervari r

on r.IDExpozitie=expo.IDExpozitie

where r.DataRezervare between @DataInitiala and @DataFinala

group by expo.DataAparitie

end

****

**6. Creati o procedura pentru afisarea functiilor ale caror salarii sunt cuprinse intre doua valori date de utilizator si in care salariul net este mai mic decat jumatate din salariul brut.**

if object\_id('ProceduraSalariiInterval','P') is not null

drop procedure ProceduraSalariiInterval

go

create procedure ProceduraSalariiInterval

@valoareMinima int,

@valoareMaxima int,

@procentaj float

as

begin

select f.NumeFunctie,s.SalariuBrut,s.SalariuNet

from Functii f

inner join SalariuFunctie sf

on f.IDFunctie=sf.IDFunctie

inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where s.SalariuBrut between @valoareMinima and @valoareMaxima

intersect

select f.NumeFunctie,s.SalariuBrut,s.SalariuNet

from Functii f

inner join SalariuFunctie sf

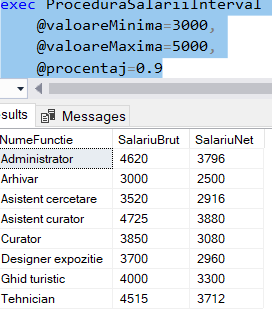
on f.IDFunctie=sf.IDFunctie

inner join Salarii s

on s.IDSalariu=sf.IDSalariu

where s.SalariuNet< s.SalariuBrut\*@procentaj

end

****

**7. Crearea unei proceduri stocate prin intermediul careia se insereaza pentru un angajat al carui ID este dat o noua functie si un nou salariu, unde parametrii inserarii sunt dati de utilizator.**

if object\_id('ProceduraAdaugareFunctieAngajat','P') is not null

drop procedure ProceduraAdaugareFunctieAngajat

go

create procedure ProceduraAdaugareFunctieAngajat

@SalariuBrut int,

@SalariuNet int,

@DataAvans Date,

@DataLichidare Date,

@OreSuplimentare int,

@ValoareAvans int,

@NumeFunctie nvarchar(40),

@IntervalOrar nvarchar(15),

@Responsabilitati nvarchar(100),

@Executie varchar(3),

@DataScoatereLaConcurs Date

as

begin

declare @idFunctie int;

declare @idSalariu int;

set @idFunctie = (select top(1) IDFunctie

from Functii

order by IDFunctie desc) +1

set @idSalariu= (select top(1) IDSalariu

from Salarii

order by IDSalariu desc) +1

insert into Functii(NumeFunctie,IntervalOrar,Responsabilitati,Executie,DataScoatereLaConcurs)

values(@NumeFunctie,@IntervalOrar,@Responsabilitati,@Executie,@DataScoatereLaConcurs)

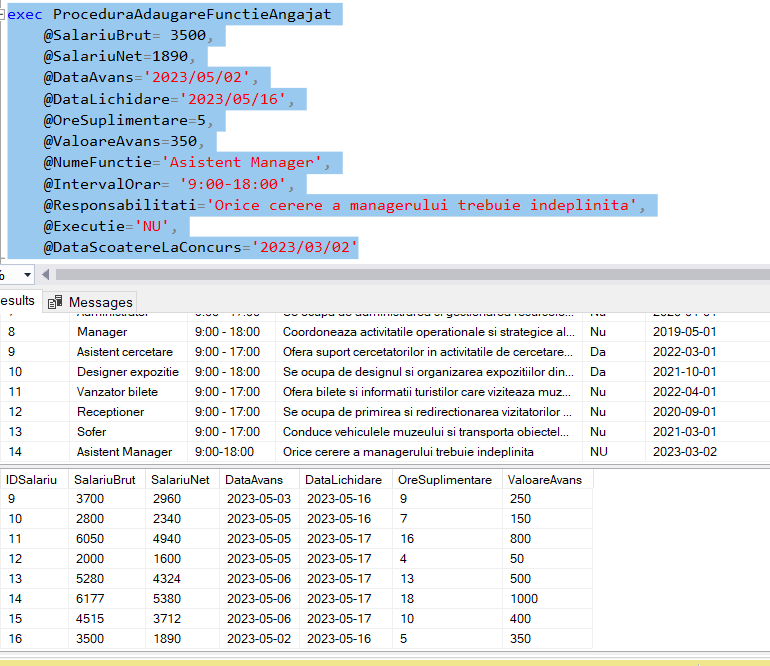
insert into Salarii(SalariuBrut,SalariuNet,DataAvans,DataLichidare,OreSuplimentare,ValoareAvans)

values (@SalariuBrut,@SalariuNet,@DataAvans,@DataLichidare,@OreSuplimentare,@ValoareAvans)

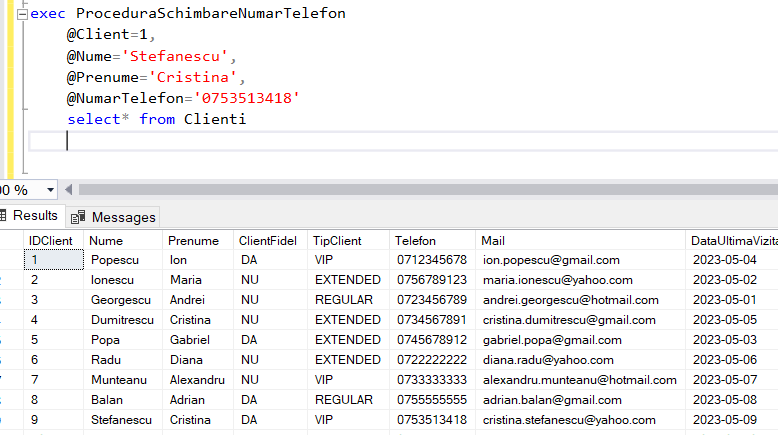
insert into SalariuFunctie(IDSalariu,IDFunctie)

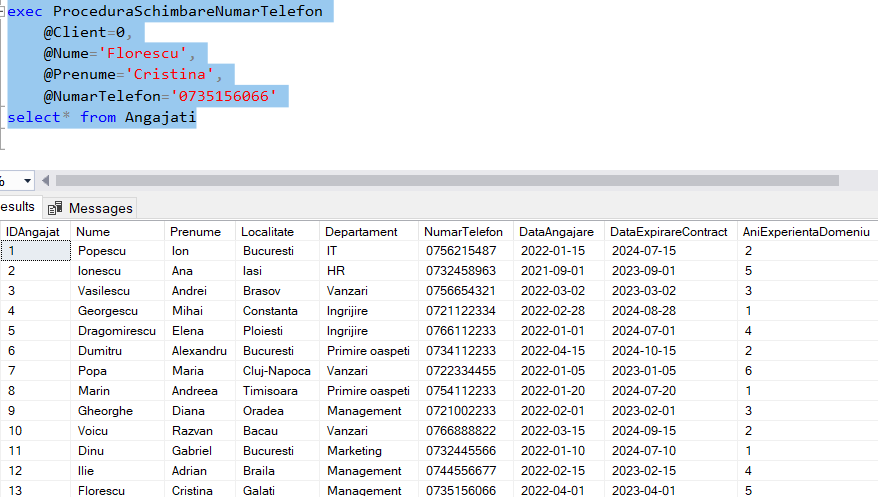
values(@idSalariu,@idFunctie)

end

****

**8. Sa se creeze o procedura stocata care sa actualizeze numarul de telefon al unui client sau al unui angajat la alegere prin initializarea unei variabile specifice(1 pentru client,restul pentru angajati) avand ca parametrii numele si prenumele acestuia, dar si numarul de telefon.**

****

****

**9. Sa se creeze o procedura stocata care sa afiseze functiile din cadrul muzeului care au un numar dat de angajati.**

if object\_id('ProceduraNumarAngajatiFunctie','P') is not null

drop procedure ProceduraNumarAngajatiFunctie

go

create procedure ProceduraNumarAngajatiFunctie

@numarAngajati int

as

begin

select f.IDFunctie,f.NumeFunctie,count(fu.IDFunctie) as NumarAngajati

from Angajati a

inner join FunctieAngajat fu

on a.IDAngajat=fu.IDAngajat

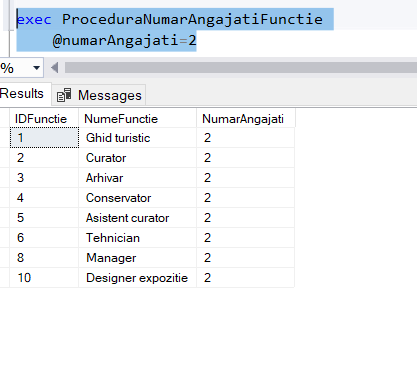
inner join Functii f

on fu.IDFunctie=f.IDFunctie

group by f.IDFunctie,f.NumeFunctie

having count(fu.IDFunctie)=@numarAngajati

end

****

**10. Crearea unei proceduri stocate care sa afiseze toti Creatori si tipurile preferate ale acestora pentru expozitia unui angajat al carui ID este dat de catre utilizatorul bazei de date.**

if object\_id('ProceduraCreatoriAngajati','P') is not null

drop procedure ProceduraCreatoriAngajati

go

create procedure ProceduraCreatoriAngajati

@IDAngajat int

as

begin

select c.Nume,c.Prenume,c.TipuriPreferate

from Angajati a

inner join Expozitii ex

on a.IDAngajat=ex.IDAngajat

inner join Exponate e

on e.IDExpozitie=ex.IDExpozitie

inner join CreatorExponat ce

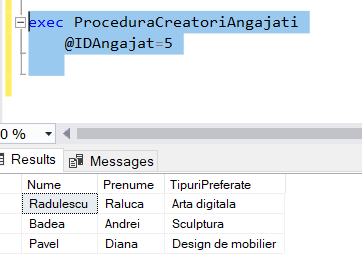
on ce.IDExponat=e.IDExponat

inner join Creatori c

on ce.IDCreator=c.IDCreator

where a.IDAngajat=@IDAngajat

end



## 3.3. Trigger-e

**1. Sa se creeze un trigger care sa nu permita inserarea unui Furnizor cu un codSRL deja existent in baza de date.**

if object\_id('TriggerDuplicatFurnizori','TR') is not null

drop trigger TriggerDuplicatFurnizori

go

create trigger TriggerDuplicatFurnizori

on Furnizori

after insert,update

as

begin

if @@ROWCOUNT =0 RETURN;

set nocount on;

if exists( select count(\*)

from Inserted as I

inner join Furnizori f

on f.CodSRL=I.CodSRL

group by I.CodSRL

having count(\*)>1

)

begin

throw 5000,'codSRL unic deja existent la nivelul Tabelei',3

end

else

begin

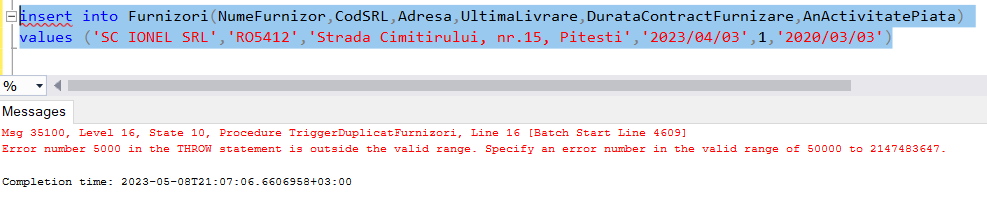
insert into Furnizori(NumeFurnizor,CodSRL,Adresa,UltimaLivrare,DurataContractFurnizare,AnActivitatePiata)

select NumeFurnizor,CodSRL,Adresa,UltimaLivrare,DurataContractFurnizare,AnActivitatePiata

from Inserted

end

end

****

**2. Sa se creeze un trigger care sa ofere un mesaj de informare referitor la introducerea cu success a unui nou Client in baza de date, in conditiile in care trebuie sa difere cel putin o informatie fata de cele ale celorlalti client pentru a fi considerat valid.**

IF OBJECT\_ID('TInsertClient', 'TR') IS NOT NULL

DROP TRIGGER TInsertClient

GO

CREATE TRIGGER TInsertClient

ON Clienti

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS(SELECT \*

FROM Clienti c

inner join Inserted i

on i.Nume=c.Nume

WHERE c.Nume =i.Nume and c.Prenume= i.Prenume and c.Telefon=i.Telefon

)

BEGIN

print 'Insert Nereusit.Userul exista deja!';

throw 5234,'Eroare Insert',3

END

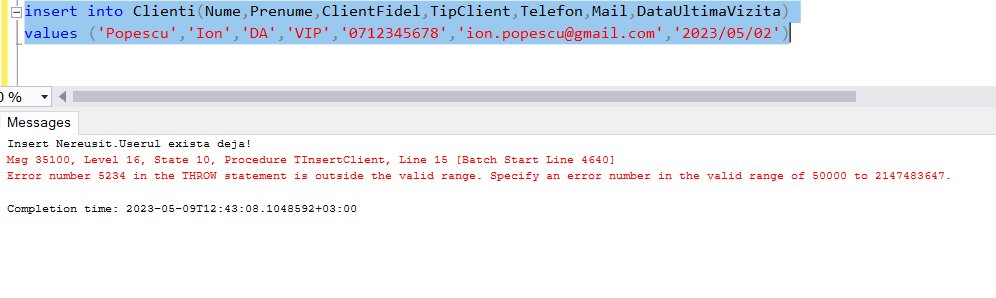
else

begin

print 'Insert reusit!'

end

END

****

**3. Sa se creeze un trigger care sa afiseze un mesaj de eroare in cazul in care se incearca o stergere din baza de date a unui utilizator care are functia de Manager.**

IF OBJECT\_ID('TDeleteAngajati', 'TR') IS NOT NULL

DROP TRIGGER TDeleteAngajati

GO

CREATE TRIGGER TDeleteAngajati

ON FunctieAngajat

AFTER DELETE,UPDATE

AS

BEGIN

SET NOCOUNT ON;

IF EXISTS(

select\*

from Deleted d

inner join Functii f

on d.IDFunctie=f.IDFunctie

where f.NumeFunctie='Manager'

)

BEGIN

print 'Nu s-a putut efectua stergerea. Angajatul ocupa functia de manager!';

throw 5234,'Eroare Delete',3

END

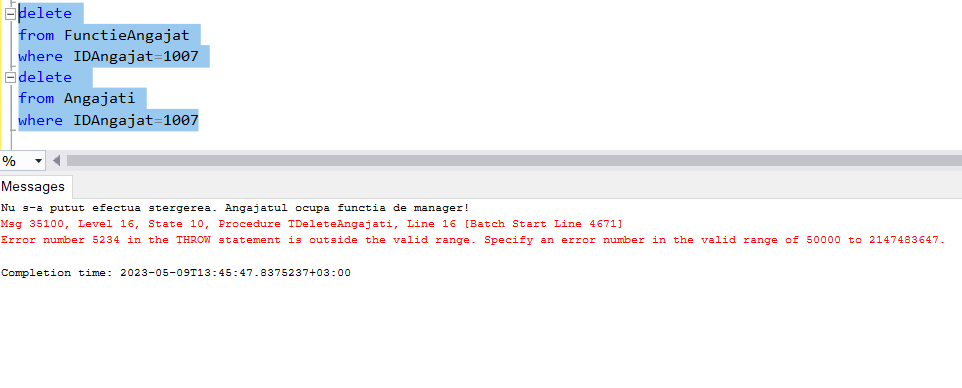
else

begin

print 'Stergere reusita!'

end

END4.

****

**4. Sa se creeze un trigger care sa nu permita introducerea unui furnizor al carui cod SRL sa nu contina initial codul RO.**

IF OBJECT\_ID('TInsertFurnizor', 'TR') IS NOT NULL

DROP TRIGGER TInsertFurnizor

GO

CREATE TRIGGER TInsertFurnizor

ON Furnizori

AFTER INSERT,UPDATE

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

IF EXISTS(

select\*

from Inserted

where CodSRL not like 'RO%'

)

BEGIN

print 'Codul SRL al Furnizorului trebuie sa aiba prefixul RO';

throw 5234,'Eroare Delete',3

END

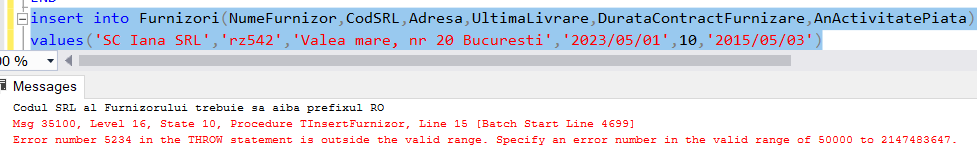
else

begin

print 'Introducere reusita!'

end

END

****

**5. Sa se creeze un trigger care sa afiseze, la punerea in posesie a unei unelte unui angajat din cadrul muzeului , numarul de unelte pe care acesta il are pe inventar.**

IF OBJECT\_ID('TPosesieUnealta', 'TR') IS NOT NULL

DROP TRIGGER TPosesieUnealta

GO

CREATE TRIGGER TPosesieUnealta

ON AngajatUnealta

AFTER INSERT

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

begin

declare @numarUnelte int;

declare @numeUnealta nvarchar(30);

declare @prenumeAngajat nvarchar(30);

declare @numeAngajat nvarchar(30);

set @numarUnelte=(select count(\*)

from inserted i

inner join AngajatUnealta au

on au.IDAngajat=i.IDAngajat

inner join Unelte u

on u.IDUnealta=au.IDUnealta

group by i.IDAngajat

)

set @numeAngajat= (select a.Nume

from Inserted i

inner join Angajati a

on i.IDAngajat=a.IDAngajat

);

set @prenumeAngajat=(select a.Prenume

from Inserted i

inner join Angajati a

on i.IDAngajat=a.IDAngajat);

set @numeUnealta=(select u.Nume

from Inserted i

inner join Unelte u

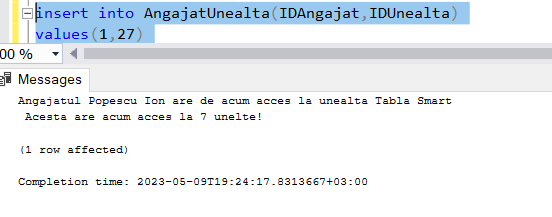
on i.IDUnealta=u.IDUnealta);

print 'Angajatul '+ @numeAngajat + ' '+@prenumeAngajat+' are de acum acces la unealta ' +@numeUnealta

print ' Acesta are acum acces la ' + cast((@numarUnelte+1) as nvarchar(50))+' unelte!'

end

END

****

**6. Sa se creeze un trigger care sa afiseze un mesaj specific atunci cand unui angajat ii este acordata o functie cu un salariu peste 2500 de lei net.**

IF OBJECT\_ID('TInsertAngajatFunctie', 'TR') IS NOT NULL

DROP TRIGGER TInsertAngajatFuctie

GO

CREATE TRIGGER TInsertAngajatFunctie

ON FunctieAngajat

AFTER INSERT

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

BEGIN

declare @SalariuFunctie int;

set @SalariuFunctie=(select s.SalariuNet

from Inserted i

inner join Functii f

on f.IDFunctie=i.IDFunctie

inner join SalariuFunctie sf

on sf.IDFunctie=f.IDFunctie

inner join Salarii s

on s.IDSalariu=sf.IDSalariu)

if @SalariuFunctie<2500

begin

print 'Salariul nu este mai mare de 2500 de lei';

end

else

begin

print 'Salariul este mai mare de 2500 de lei';

end

END

****

**7. Sa se creeze un trigger care sa interzica introducerea unei date de expirare a contractului unui angajat mai mica decat cea de angajare a acestuia.**

IF OBJECT\_ID('TInsertDataAngajati', 'TR') IS NOT NULL

DROP TRIGGER TInsertDataAngajati

GO

CREATE TRIGGER TInsertDataAngajati

ON Angajati

AFTER INSERT,UPDATE

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

if (select DataAngajare

from inserted)> (select DataExpirareContract

from inserted)

begin

print 'Eroare fatala, DataExpirareContract trebuie sa fie mai mare decat DataAngajare';

throw 5234,'Eroare fatala!',3

end

else

begin

print 'Insert realizat cu succes!'

end

END

****

**8. Sa se creeze un trigger care sa ofere un mesaj de eroare si sa nu permita adaugarea a mai mult de 4 suveniruri de tip magnet din Romania.**

IF OBJECT\_ID('TInsertSuvenirMagnetRomania', 'TR') IS NOT NULL

DROP TRIGGER TInsertSuvenirMagnetRomania

GO

CREATE TRIGGER TInsertSuvenirMagnetRomania

ON Suveniruri

AFTER INSERT,UPDATE

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

declare @numarMagneti int;

set @numarMagneti= (select count(\*)

from Suveniruri

where Tip='Magnet' and TaraProvenienta='Romania'

group by Tip,TaraProvenienta)

if @numarMagneti>=4

begin

print 'Eroare fatala! Nu pot fi introdusi mai mult de 4 magneti din Romania';

throw 5234,'Eroare',3

end

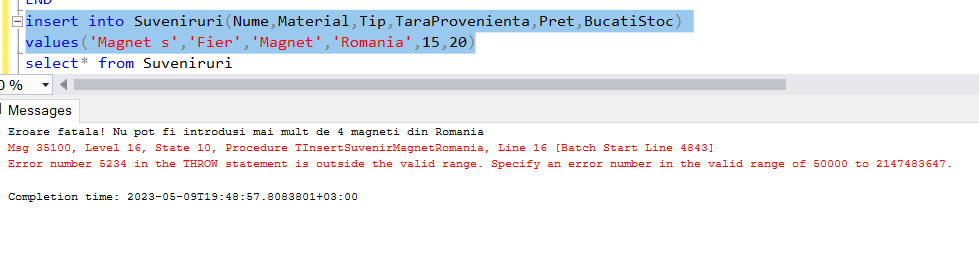
else

begin

print 'Insert realizat cu succes!'

end

END

****

**9. Sa se creeze un trigger care sa nu permita marirea salariului brut al unui angajat cu o suma mai mare de 1000 de lei.**

IF OBJECT\_ID('TMarireSalariu', 'TR') IS NOT NULL

DROP TRIGGER TMarireSalariu

GO

CREATE TRIGGER TMarireSalariu

ON Salarii

AFTER UPDATE

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

if exists(select\*

from Salarii s

inner join inserted i

on i.IDSalariu=s.IDSalariu

where i.SalariuBrut-s.SalariuBrut<1000

)

begin

print 'Nu poate exista o astfel de crestere salariala!';

throw 5234,'Eroare',2

end

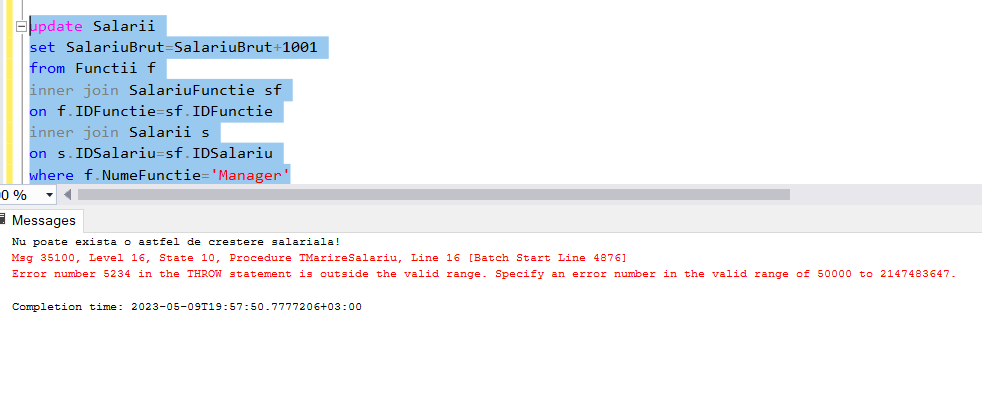
else

begin

print 'Marire de salariu realizata cu succes!'

end

END

****

**10. Sa se creeze un trigger care sa afiseze un mesaj sugestiv in cazul in care este modificata grila salariala a unui angajat care are id par.**

IF OBJECT\_ID('TModificareGrila', 'TR') IS NOT NULL

DROP TRIGGER TModificareGrila

GO

CREATE TRIGGER TModificareGrila

ON SalariuFunctie

AFTER Insert

AS

BEGIN

if @@ROWCOUNT=0 RETURN;

SET NOCOUNT ON;

if exists(

select \*

from Angajati a

inner join FunctieAngajat fa

on a.IDAngajat=fa.IDAngajat

inner join Functii f

on f.IDFunctie=fa.IDFunctie

inner join inserted i

on i.IDFunctie=f.IDFunctie

inner join Salarii s

on s.IDSalariu=i.IDSalariu

where a.IDAngajat%2=0

)

begin

print 'Unui angajat cu id-ul numar par i s-a modificat grila salariala!'

end

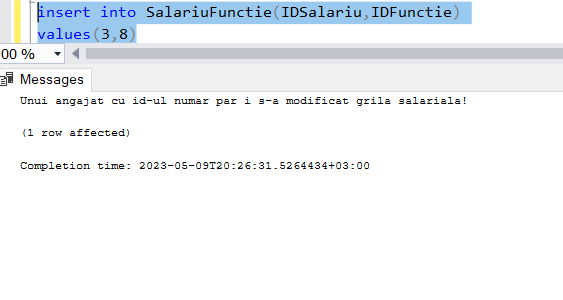
else

begin

print 'Nu s-a modificat grila salariala a unui angajat cu id-ul par!'

end

END

****

**4. Centralizare**

* Cerințe care implementează interogări SELECT: 63
* Cerințe care implementează interogări INSERT: 21
* Cerințe care implementează interogări UPDATE: 20
* Cerințe care implementează interogări DELETE: 15
* Cerințe care implementează expresii de tabel comune sau CTE: 5
* Cerințe care implementează vizualizări sau VIEW: 7
* Cerințe care implementează tranzacții sau TRANSACTIONS: 10
* Cerințe care implementează procedure stocate sau STORED PROCEDURES: 10
* Cerințe care implementează triggere sau TRIGGERS: 10
* TOTAL CERINȚE: 163