**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Programul de studii: Tehnologia informației**

RAPORT

# la Baze de Date

**Lucrare individuală II BD-SQL**

**Tema: Creare bază de date**

A efectuat:

st. gr. TI-211 Popa Cătălin

A verificat: Olga Grosu,

Saranciuc Dorian

UTM, Chișinău 2023

**Cuprins**

1. Schema bazei de date...........................................................................................................3
2. Crearea tabelelor in MS SQL Server....................................................................................4
3. Popularea cu tupluri a bazei de date......................................................................................6
4. Interogări asupra bazei de date...........................................................................................12
5. Concluzii............................................................................................................................19
6. Anexa 1 – Summary...........................................................................................................21
7. Anexa 2 – Scriptul SQL unic..............................................................................................22

**1. Schema bazei de date**

Schema bazei de date pe care am creat-o, reprezintă relațiile care au loc între un bon de la magazin și restul datelor care depind de el. Schema este reprezentată în figura 1.

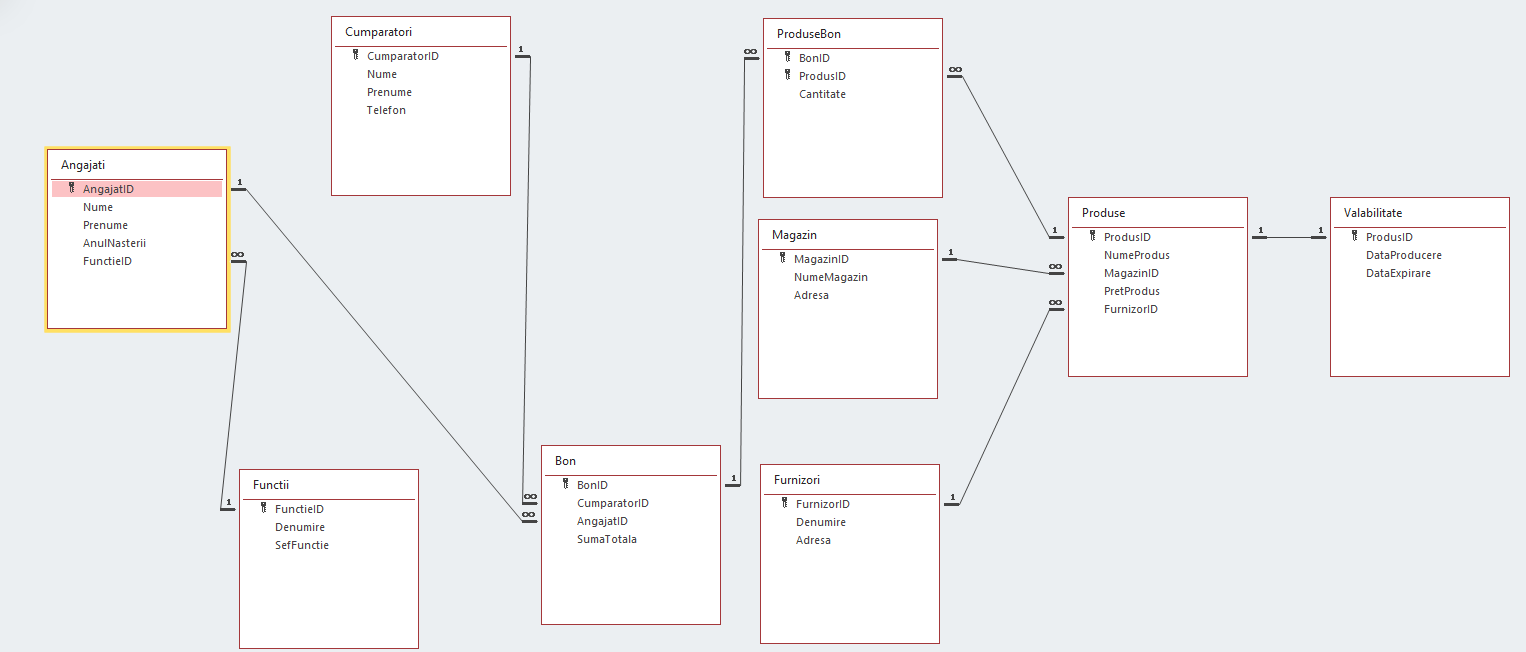


Figura 1 – Schema bazei de date.

Baza de date "LucruIndividual" este concepută pentru gestionarea informațiilor legate de un sistem de magazin, incluzând date despre magazine, furnizori, cumpărători, funcții și angajați, produse, valabilitatea acestora, bonuri și produsele asociate acestora.

După cum vedem, baza de date este creată din 9 tabele și toate au careva relații între ele. În baza de date sunt sunt toate trei relații, precum unu la unu între tabela Produse și Valabilitate. Relația unu la mulți între tabela funcții și angajați și alte tabele. Relația mulți la mulți care este realizată prin trei tabele Bon, ProduseBon și Produse. Tabela "Magazin" conține informații despre diferite magazine, cum ar fi numele și adresa lor. Tabela "Furnizori" cuprinde detalii despre furnizorii de produse, inclusiv denumire și adresa. Tabela "Cumparatori" stochează informații despre clienți, cum ar fi numele, prenumele și numărul de telefon.Tabelele "Functii" și "Angajati" sunt concepute pentru gestionarea informațiilor despre funcțiile angajaților și datele personale ale acestora, precum și detalii despre anul nașterii. O funcționalitate interesantă este adăugarea unei coloane calculate "NumeComplet" în tabela "Angajati".Tabela "Produse" conține informații despre produsele disponibile în magazine, inclusiv nume, preț, furnizor și magazin asociat. Tabela "Valabilitate" gestionează informații legate de perioada de valabilitate a produselor.Tabelele "Bon" și "ProduseBon" sunt interconectate, permițând urmărirea achizițiilor cumpărătorilor și a produselor achiziționate în cadrul unui bon, inclusiv cantitatea fiecărui produs și suma totală a bonului.În general, această structură de bază de date oferă o fundație solidă pentru gestionarea informațiilor legate de activitatea unui lanț de magazine, inclusiv detaliile despre produse, angajați, clienți și tranzacții comerciale.

**2. Crearea tabelelor in MS SQL Server**

După realizarea schemei, am creat tabelele și baza de date în Microsoft SQL Server Management Studio. În primul rând verificăm dacă baza de date există, iar apoi creăm tabelele.

IF EXISTS (SELECT \* FROM sys.databases WHERE name = 'SistemGestionalBon')

BEGIN

USE master;

DROP DATABASE SistemGestionalBon;

END;

CREATE DATABASE SistemGestionalBon;

Mai jos sunt reprezentate interogările de creare a tuturor tabelelor.

-- Creare tabela Magazin

CREATE TABLE SistemGestionalBon.dbo.Magazin (

MagazinID TINYINT IDENTITY(1,1) PRIMARY KEY,

NumeMagazin VARCHAR(255),

Adresa VARCHAR(255)

);

-- Creare tabela Furnizori

CREATE TABLE SistemGestionalBon.dbo.Furnizori (

FurnizorID SMALLINT PRIMARY KEY,

Denumire VARCHAR(255),

Adresa VARCHAR(255)

);

-- Creare tabela Cumparatori

CREATE TABLE SistemGestionalBon.dbo.Cumparatori (

CumparatorID INT PRIMARY KEY,

Nume VARCHAR(255),

Prenume VARCHAR(255),

Telefon VARCHAR(15)

);

-- Creare tabela Functii

CREATE TABLE SistemGestionalBon.dbo.Functii (

FunctieID INT PRIMARY KEY,

Denumire VARCHAR(255),

SefFunctie VARCHAR(255)

);

-- Creare tabela Angajati

CREATE TABLE SistemGestionalBon.dbo.Angajati (

AngajatID SMALLINT PRIMARY KEY,

Nume VARCHAR(255),

Prenume VARCHAR(255),

AnulNasterii INT,

FunctieID INT,

NumeComplet AS (Nume + ' ' + Prenume) PERSISTED

FOREIGN KEY (FunctieID) REFERENCES Functii(FunctieID)

);

-- Creare tabela Produse

CREATE TABLE SistemGestionalBon.dbo.Produse (

ProdusID INT PRIMARY KEY,

NumeProdus VARCHAR(255),

MagazinID TINYINT,

PretProdus DECIMAL(10, 2),

FurnizorID SMALLINT,

FOREIGN KEY (MagazinID) REFERENCES Magazin(MagazinID),

FOREIGN KEY (FurnizorID) REFERENCES Furnizori(FurnizorID));

--Creare tabela valabilitate

CREATE TABLE SistemGestionalBon.dbo.Valabilitate (

ProdusID INT PRIMARY KEY,

DataProducere DATE,

DataExpirare DATE,

FOREIGN KEY (ProdusID) REFERENCES Produse(ProdusID)

);

-- Creare tabela Bon

CREATE TABLE SistemGestionalBon.dbo.Bon (

BonID INT PRIMARY KEY,

CumparatorID INT,

AngajatID SMALLINT,

SumaTotala DECIMAL(10, 2),

FOREIGN KEY (CumparatorID) REFERENCES Cumparatori(CumparatorID),

FOREIGN KEY (AngajatID) REFERENCES Angajati(AngajatID)

);

-- Creare tabela ProduseBon

CREATE TABLE SistemGestionalBon.dbo.ProduseBon (

BonID INT,

ProdusID INT,

Cantitate INT,

PRIMARY KEY (BonID, ProdusID),

FOREIGN KEY (BonID) REFERENCES Bon(BonID),

FOREIGN KEY (ProdusID) REFERENCES Produse(ProdusID)

);

În figura 2, sunt prezentate tabelele.

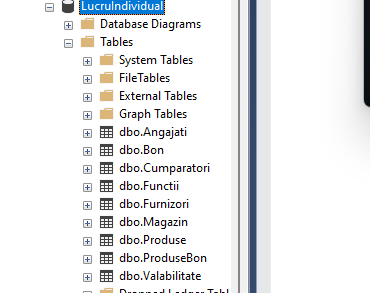


Figura 2 – Crearea tabelelor în baza ded date.

După crearea cu succes a bazei de date și tabelelor, este nevoie să completăm tabelele cu tupluri.

**3. Popularea cu tupluri a bazei de date**

-- Inserare date in tabela Angajati

INSERT INTO SistemGestionalBon.dbo.Angajati (AngajatID, Nume, Prenume, AnulNasterii, FunctieID)

VALUES

(1, 'Popescu', 'Ion', 1980, 1),

(2, 'Negura', 'Ana', 1990, 2),

(3, 'Dumitru', 'Mihai', 1985, 3),

(4, 'Gabriela', 'Vasile', 1982, 4),

(5, 'Radu', 'Elena', 1995, 5),

(6, 'Ana', 'Adrian', 1988, 6),

(7, 'Vasile', 'Cristina', 1980, 7),

(8, 'Cristian', 'Mihai', 1992, 8),

(9, 'Mihai', 'Alexandru', 1985, 9),

(10, 'Adrian', 'Laura', 1987, 10),

(11, 'Elena', 'Vlad', 1990, 11),

(12, 'Iuliana', 'Raul', 1983, 12),

(13, 'Valentin', 'Ciprian', 1986, 13),

(14, 'Popa', 'Galina', 1979, 11),

(15, 'Grigorenco', 'Ion', 1899, 13);

Tabela Angajați este afișată în figura 3. În ea avem coloanele AngajatID, Nume, Prenume, AnulNasterii și FunctieID.

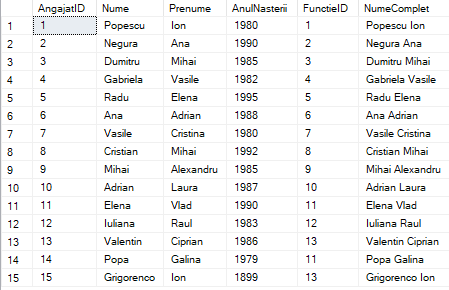


Figura 3 – Tabela Angajați.

-- Inserare date in tabela Bon

INSERT INTO SistemGestionalBon.dbo.Bon (BonID, CumparatorID, AngajatID, SumaTotala)

VALUES

(1, 1, 1, 110.00),

(2, 2, 2, 75.00),

(3, 3, 13, 50.00),

(4, 2, 2, 130.00),

(5, 5, 3, 90.00),

(6, 13, 10, 115.00),

(7, 8, 2, 78.00),

(8, 9, 9, 55.00),

(9, 8, 1, 75.00),

(10, 6, 2, 105.00),

(11, 3, 2, 68.00),

(12, 13, 2, 40.00),

(13, 12, 6, 50.00),

(14, 15, 5, 12),(15, 3, 1, 100);

Tabela Bon este afișată în figura 4. Ea conține coloanele BonID, CumparatorID, AngajatID și SumaTotala.

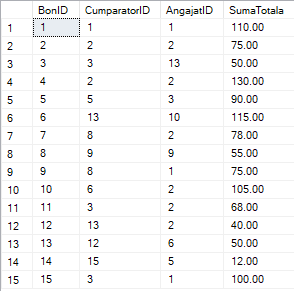


Figura 4 – Tabela Bon.

-- Inserare date in tabela Cumparatori

INSERT INTO SistemGestionalBon.dbo.Cumparatori (CumparatorID, Nume, Prenume, Telefon)

VALUES

(1, 'Ion', 'Popescu', '062048221'),

(2, 'Ana', 'Ionescu', '079342896'),

(3, 'Mihai', 'Georgescu', '062097521'),

(4, 'Elena', 'Dumitru', '062048982'),

(5, 'Adrian', 'Popa', '062918221'),

(6, 'Cristina', 'Vasilescu', '069248221'),

(7, 'George', 'Stancu', '062872121'),

(8, 'Andreea', 'Munteanu', '062049989'),

(9, 'Alex', 'Constantinescu', '069898221'),

(10, 'Laura', 'Moldovan', '062098321'),

(11, 'Vlad', 'Alexandrescu', '068676221'),

(12, 'Raluca', 'Radu', '062088221'),

(13, 'Ciprian', 'Iancu', '062023211'),

(14, 'Catalin', 'Popa', '062028456'),

(15, 'Ion', 'Mardarenco', '079325323');

Tabela Cumpărători este afișată în figura 5. Ea conține coloanele CumparatorID, Nume, Prenume și Telefon.



Figura 5 – Tabela Cumpărători.

-- Inserare date in tabela Functii

INSERT INTO SistemGestionalBon.dbo.Functii (FunctieID, Denumire, SefFunctie)

VALUES

(1, 'Manager', 'Marius Popescu'),

(2, 'Vanzator', 'Andreea Ionescu'),

(3, 'Casier', 'George Dumitrescu'),

(4, 'Receptioner', 'Marius Popescu'),

(5, 'Vanzator secundar', 'Mihai Patic'),

(6, 'Casier principal', 'George Petrache'),

(7, 'Logistician', 'Marius Popescu'),

(8, 'Vanzator principal', 'Cristina Ioana'),

(9, 'Hamal', 'George Ionescu'),

(10, 'Supraveghetor', 'Marius Popescu'),

(11, 'Consultant', 'Catalin Popa'),

(12, 'Sofer', 'George Patic'),

(13, 'Paznic', 'Alexandru Chirica');

Tabela Funcții este afișată în figura 6. Ea conține coloanele FunctieID, Denumire și SefFunctie.



Figura 6 – Tabela Funcții.

-- Inserare date in tabela Furnizori

INSERT INTO SistemGestionalBon.dbo.Furnizori (FurnizorID, Denumire, Adresa)

VALUES

(1, 'Flroare', 'Strada Fagilor, nr. 10'),

(2, 'Carpet', 'Aleea Florilor, nr. 25'),

(3, 'Alimente SRL', 'Bulevardul Rozelor, nr. 40'),

(4, 'Delicios Foods SA', 'Aleea Miresmeilor, nr. 15'),

(5, 'Fresco Distributie Alimentara', 'Strada Perlelor, nr. 30'),

(6, 'BioFarm Produse Organice', 'Bulevardul Colibelor, nr. 45'),

(7, 'Gusturi Traditionale Import-Export', 'Aleea Bijuteriilor, nr. 60'),

(8, 'Marele Pantry Importer', 'Strada Elegantei, nr. 75'),

(9, 'ProNatura Distributie', 'Bulevardul Refined, nr. 90'),

(10, 'Alimex Companie Alimentara', 'Aleea Preciosilor, nr. 105'),

(11, 'EcoBio Distributie Bio', 'Strada Eterna, nr. 120'),

(12, 'MegaFoods Importer', 'Bulevardul Modernitatii, nr. 135'),

(13, 'TopGusturi Distributie', 'Aleea Impecabililor, nr. 150'),

(14, 'ProdusLocal SA', 'Dacia, nr32'),

(15, 'ImportExport', 'Independentei, nr 322');

Tabela furnizori este afișată în figura 7. Ea conține coloanele FurnizorID, Denumire și Adresa.



Figura 7 – Tabela Furnizori.

-- Inserare date in tabela Magazin

INSERT INTO SistemGestionalBon.dbo.Magazin (NumeMagazin, Adresa)

VALUES

('Mega Store', 'Strada Primaverii, nr. 123'),

('SuperMart', 'Bulevardul Victoriei, nr. 456'),

('Fresh Market', 'Aleea Fructelor, nr. 789'),

('Bonus', 'Bulevardul Unirii, nr. 789'),

('Nr1', 'Strada Luminii, nr. 456'),

('Linella', 'Aleea Viselor, nr. 123'),

('Kaufland', 'Strada Soarelui, nr. 101'),

('Local', 'Bulevardul Lunii, nr. 222'),

('Metro', 'Aleea Stelelor, nr. 333'),

('City Supermarket', 'Strada Norilor, nr. 444'),

('Mall', 'Bulevardul Ploilor, nr. 555'),

('Sun City', 'Aleea Fulgilor, nr. 666'),

('Blue Sky Market', 'Strada Vantului, nr. 777'),

('Nr2', 'Sarmisegetusa, nr.43'),

('Metro 2', 'Dacia, nr21');

Tabela Magazin este afișată în figura 8. Ea conține coloanele MagazinID, NumeMagazin și Adresa.



Figura 8 – Tabela Magazin.

-- Inserare date in tabela Produse

INSERT INTO SistemGestionalBon.dbo.Produse (ProdusID, NumeProdus, MagazinID, PretProdus, FurnizorID)

VALUES

(1, 'Făină integrală bio', 1, 50.00, 1),

(2, 'Ovăz ecologic', 4, 30.00, 2),

(3, 'Miere de albine pură', 3, 25.00, 3),

(4, 'Quinoa organica', 7, 40.00, 4),

(5, 'Ulei de măsline extravirgin', 5, 28.00, 5),

(6, 'Nuci organice', 8, 22.00, 6),

(7, 'Ciocolată neagră bio', 9, 60.00, 7),

(8, 'Fructe uscate mix', 9, 35.00, 8),

(9, 'Ceai verde organic', 9, 18.00, 9),

(10, 'Mazăre congelată', 10, 45.00, 10),

(11, 'Oțet balsamic premium', 11, 32.00, 11),

(12, 'Lapte de migdale', 12, 25.00, 12),

(13, 'Sos de roșii organic', 8, 55.00, 13),

(14, 'Peste rosu', 13, 52, 4),

(15, 'Apa minerala', 4, 12, 2);

Tabela Produse este afișată în figura 9. Ea conține coloanele ProdusID, NumeProdus, MagazinID, PretProdus și FurnizorID.

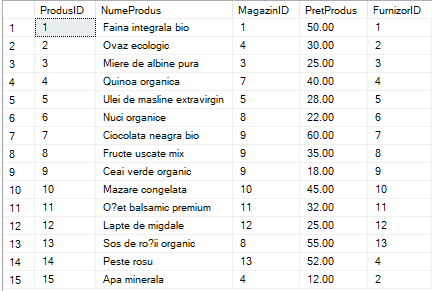


Figura 9 – Tabela Produse.

-- Inserare date in tabela ProduseBon

INSERT INTO SistemGestionalBon.dbo.ProduseBon (BonID, ProdusID, Cantitate)

VALUES

(1, 1, 2),

(1, 2, 1),

(2, 3, 3),

(2, 2, 2),

(3, 1, 1),

(3, 3, 2),

(4, 4, 3),

(4, 5, 1),

(5, 6, 2),

(5, 7, 1),

(6, 8, 3),

(6, 9, 1),

(7, 10, 2);

Tabela ProduseBon este afișată în figura 10. Conține coloanele BonID, ProdusID și Cantitate.

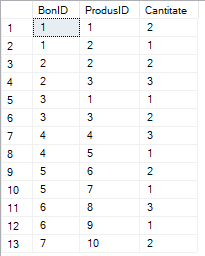


Figura 10 – Tabela ProduseBon.

-- Inserare date in tabela Valabilitate

INSERT INTO SistemGestionalBon.dbo.Valabilitate (ProdusID, DataProducere, DataExpirare)

VALUES

(1, '2022-01-01', '2022-12-31'),

(2, '2021-05-15', '2022-05-15'),

(3, '2022-03-10', '2023-03-10'),

(4, '2021-11-15', '2022-11-15'),

(5, '2022-06-20', '2023-06-20'),

(6, '2021-08-05', '2022-08-05'),

(7, '2022-02-28', '2023-02-28'),

(8, '2021-12-10', '2022-12-10'),

(9, '2022-04-25', '2023-04-25'),

(10, '2022-07-01', '2023-07-01'),

(11, '2021-09-12', '2022-09-12'),

(12, '2022-01-05', '2023-01-05'),

(13, '2022-10-30', '2023-10-30'),

(14, '2022-11-30', '2024-10-30'),

(15, '2023-12-02', '2024-10-30');

Tabela Valabilitate este afilată în figura 11. Ea conține coloanele ProdusID, DataProducere și DataExpirare.

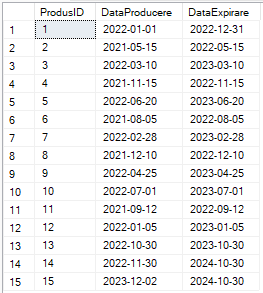


Figura 10 – Tabela Valabilitate.

**4. Interogări asupra bazei de date**

--va afisa produsul si denumirea furnizorului

SELECT a.NumeProdus, b.Denumire as furnizor

FROM SistemGestionalBon.dbo.Produse a

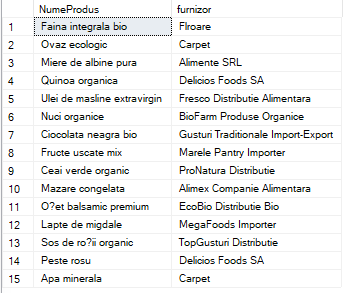
INNER JOIN SistemGestionalBon.dbo.Furnizori b ON a.FurnizorID = b.FurnizorID; 

Figura 11 – Inner Join.

--va afisa cumaraotrii si suma totala a bonuriloe

SELECT Cumparatori.Nume, Cumparatori.Prenume, Bon.SumaTotala

FROM SistemGestionalBon.dbo.Cumparatori

LEFT JOIN SistemGestionalBon.dbo.Bon ON Cumparatori.CumparatorID = Bon.CumparatorID;



Figura 12 – Left Join.

--Afișează denumirea funcțiilor și numele complet al angajaților. Funcțiile fără angajați vor apărea și ele în rezultat.

SELECT Functii.Denumire, Angajati.NumeComplet

FROM SistemGestionalBon.dbo.Functii

RIGHT JOIN SistemGestionalBon.dbo.Angajati ON Functii.FunctieID = Angajati.FunctieID; 

Figura 13 – Right Join.

--Afișează denumirea funcțiilor și numele complet al angajaților. Funcțiile fără angajați vor apărea și ele în rezultat.

SELECT Functii.Denumire, Angajati.NumeComplet

FROM SistemGestionalBon.dbo.Functii

RIGHT JOIN SistemGestionalBon.dbo.Angajati ON Functii.FunctieID = Angajati.FunctieID;



Figura 14 – Right Join.

--Afișează numele magazinelor, numele produselor și prețurile produselor pentru acele produse care au un preț mai mare de 30.00.

SELECT Magazin.NumeMagazin, Produse.NumeProdus, Produse.PretProdus

FROM SistemGestionalBon.dbo.Magazin

INNER JOIN SistemGestionalBon.dbo.Produse ON Magazin.MagazinID = Produse.MagazinID

WHERE Produse.PretProdus > 30.00;

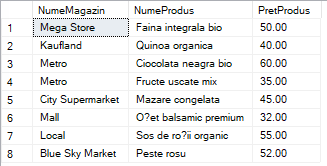


Figura 15 – Inner join si filtrare.

--Afișează numele complet al angajaților, denumirea funcțiilor și suma totală a bonurilor asociate.

SELECT Angajati.NumeComplet, Functii.Denumire, sum(Bon.SumaTotala)

FROM SistemGestionalBon.dbo.Angajati

INNER JOIN SistemGestionalBon.dbo.Functii ON Angajati.FunctieID = Functii.FunctieID

INNER JOIN SistemGestionalBon.dbo.Bon ON Angajati.AngajatID = Bon.AngajatID

group by Angajati.NumeComplet, Functii.Denumire;

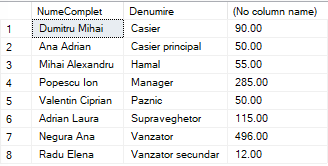


Figura 16 – Inner join și group by.

--Afișează numele și prenumele cumpărătorilor care au un număr de telefon care începe cu '062'.

SELECT Nume, Prenume, Telefon

FROM SistemGestionalBon.dbo.Cumparatori

WHERE Telefon LIKE '062%';

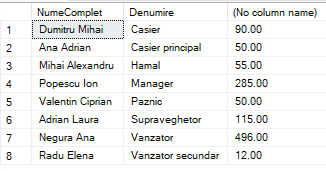


Figura 17 – Filtru cu like.

--Afișează media duratei de valabilitate (în zile) a produselor din tabela de valabilitate.

SELECT AVG(DATEDIFF(day, DataProducere, DataExpirare)) AS MedieDurataValabilitate

FROM SistemGestionalBon.dbo.Valabilitate;



Figura 18 – Functia AVG.

--Afișează denumirile și adresele furnizorilor și magazinelor într-un singur set distinct de rezultate.

SELECT Denumire, Adresa FROM SistemGestionalBon.dbo.Furnizori

UNION

SELECT NumeMagazin, Adresa FROM SistemGestionalBon.dbo.Magazin;

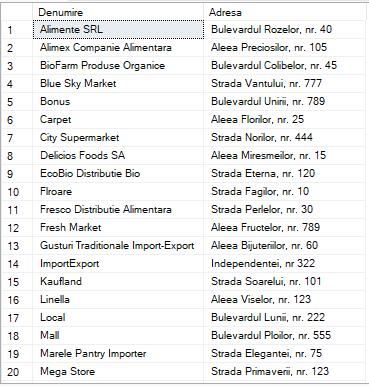


Figura 19 – UNION.

--Afișează numele complet al angajaților care au asociate bonuri în tabela Bon.

SELECT NumeComplet

FROM SistemGestionalBon.dbo.Angajati AS A

WHERE EXISTS (

SELECT 1

FROM SistemGestionalBon.dbo.Bon AS B

WHERE A.AngajatID = B.AngajatID);



Figura 20 – EXISTS.

--Afișează numărul de produse pentru fiecare bon, folosind funcția COUNT și GROUP BY.

SELECT a.BonID, COUNT(a.ProdusID) AS NumarProduse

FROM SistemGestionalBon.dbo.ProduseBon a

GROUP BY a.BonID;

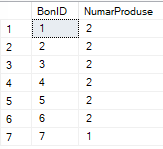


Figura 21 – Count si Group by.

--Afișează numele și prenumele cumpărătorilor care nu au asociate bonuri.

SELECT Cumparatori.Nume, Cumparatori.Prenume

FROM SistemGestionalBon.dbo.Cumparatori

LEFT JOIN SistemGestionalBon.dbo.Bon ON Cumparatori.CumparatorID = Bon.CumparatorID

WHERE Bon.BonID IS NULL;

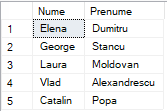


Figura 22 – Left join si filtru.

--Afișează produsul și data maximă de expirare asociată fiecărui produs.

SELECT ProdusID, MAX(DataExpirare) AS DataExpirareMaxima

FROM SistemGestionalBon.dbo.Valabilitate

GROUP BY ProdusID;

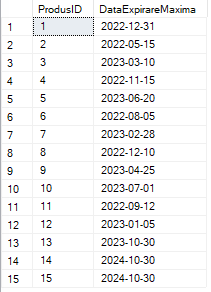


Figura 23 – Functia MAX.

--Afișează numărul total de produse cumpărate de fiecare cumpărător.

SELECT B.CumparatorID, COUNT(PB.ProdusID) AS NumarProduse

FROM SistemGestionalBon.dbo.Bon AS B

INNER JOIN SistemGestionalBon.dbo.ProduseBon AS PB ON B.BonID = PB.BonID

GROUP BY B.CumparatorID;

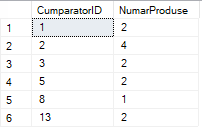


Figura 24 – Inner Join COUNT.

--Afișează numele și prețul produselor care au un preț mai mare decât media prețurilor produselor.

SELECT NumeProdus, PretProdus

FROM SistemGestionalBon.dbo.Produse

WHERE PretProdus > (SELECT AVG(PretProdus) FROM SistemGestionalBon.dbo.Produse);

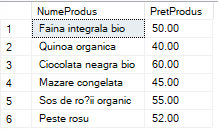


Figura 25 – AVG.

--Afișează ID-ul cumpărătorului și suma totală a cumpărăturilor pentru cei cu cumpărături mai mari de 80.00

SELECT B.CumparatorID, SUM(P.PretProdus) AS SumaTotalaCumparaturi

FROM SistemGestionalBon.dbo.Bon AS B

INNER JOIN SistemGestionalBon.dbo.ProduseBon AS PB ON B.BonID = PB.BonID

INNER JOIN SistemGestionalBon.dbo.Produse AS P ON PB.ProdusID = P.ProdusID

GROUP BY B.CumparatorID

HAVING SUM(P.PretProdus) > 80.00;



Figura 26 – HAVING.

--Afișează numele complet al angajaților și vârsta lor actuală, calculată pe baza anului de naștere.

SELECT NumeComplet, DATEPART(YEAR, GETDATE()) - AnulNasterii AS Varsta

FROM SistemGestionalBon.dbo.Angajati

where DATEPART(YEAR, GETDATE()) - AnulNasterii < 50;



Figura 27 – DATEPART.

--Afișează numele complet al angajaților folosind funcția CONCAT.

SELECT CONCAT(Nume, ' ', Prenume) AS NumeComplet

FROM SistemGestionalBon.dbo.Angajati;



Figura 28 – CONCAT.

--Afișează numele și prenumele cumpărătorilor care au asociate bonuri.

SELECT Nume, Prenume

FROM SistemGestionalBon.dbo.Cumparatori

WHERE CumparatorID IN (SELECT CumparatorID FROM SistemGestionalBon.dbo.Bon);



Figura 29 – Subselect.

--Afișează produsele și perioada de valabilitate care se află între anumite date de producere.

SELECT ProdusID, DataProducere, DataExpirare

FROM SistemGestionalBon.dbo.Valabilitate

WHERE DataProducere BETWEEN '2022-01-01' AND '2022-12-31';

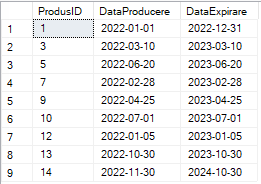


Figura 30 – BETWEEN.

**Concluzie**

În lucrarea dată, am realizat o bază de date personală, bazată pe legătura dintre bon, achitări și magazine. Baza este creată din 9 tabele și toate au careva relații între ele. În baza de date sunt sunt toate trei relații, precum unu la unu între tabela Produse și Valabilitate. Relația unu la mulți între tabela funcții și angajați și alte tabele. Relația mulți la mulți care este realizată prin trei tabele Bon, ProduseBon și Produse. Apoi am realizat mai multe interogări pe baza de date respectivă, pentru a verifica toate relațiile. Baza de date s-a obținut complexă și ușor de manipulat. Este posibilitatea de a adăuga mai multe date pentru a verifica cum va funcționa cu un volum de date mai mare.

**Anexa A (Summary)**

**Denumirea bazei de date:**

Denumirea bazei de date este "SistemGestionalBon".

**Descrierea succintă a bazei de date:**

Domeniul bazei de date este gestionarea informațiilor asociate unui sistem de magazin, inclusiv detalii despre magazine, furnizori, cumpărători, funcții și angajați, produse, valabilitatea acestora, bonuri și produsele asociate acestora.

**Screenshot din SQL Server cu diagrama bazei de date:**

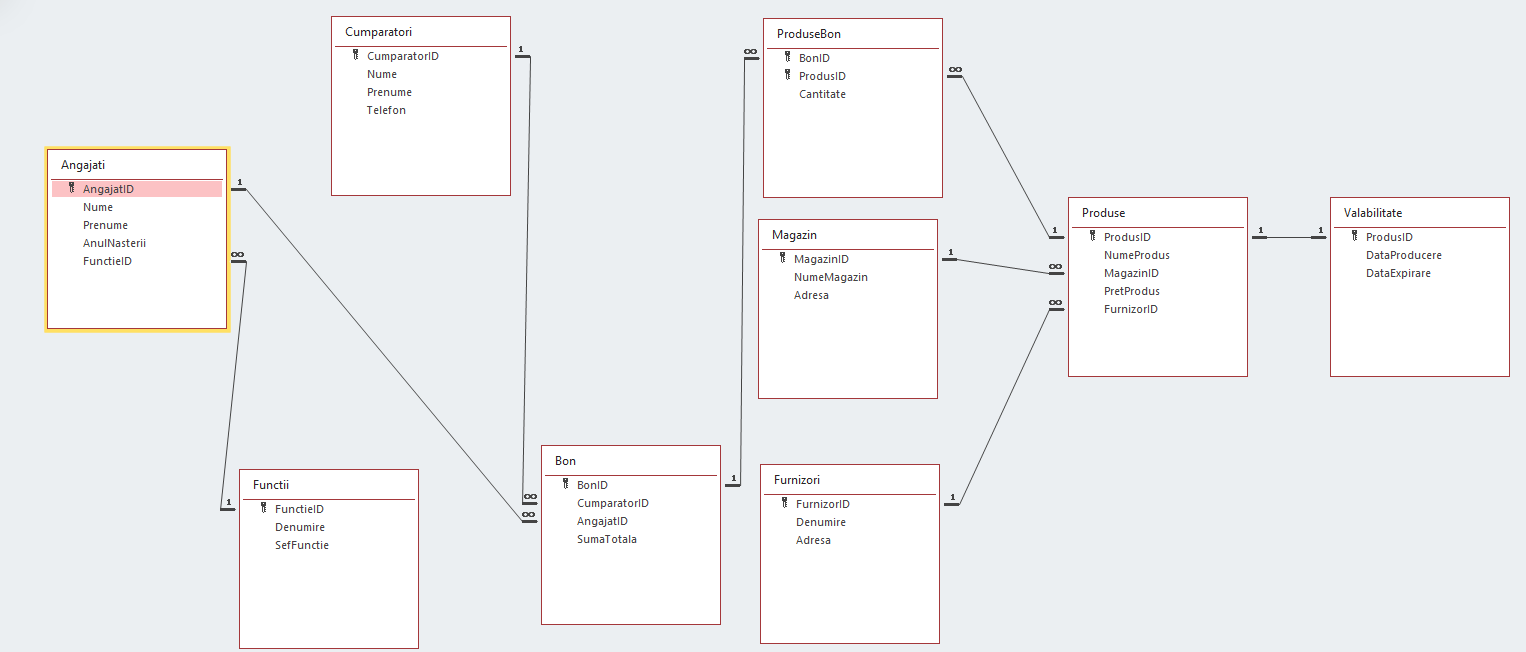


Figura 1 – Schema bazei de date.

**Schema logică a bazei de date:**

* Tabelele și atributele principale includ:
  + Tabela "Magazin" (MagazinID, NumeMagazin, Adresa).
  + Tabela "Furnizori" (FurnizorID, Denumire, Adresa).
  + Tabela "Cumparatori" (CumparatorID, Nume, Prenume, Telefon).
  + Tabela "Functii" (FunctieID, Denumire, SefFunctie).
  + Tabela "Angajati" (AngajatID, Nume, Prenume, AnulNasterii, FunctieID, NumeComplet).
  + Tabela "Produse" (ProdusID, NumeProdus, MagazinID, PretProdus, FurnizorID).
  + Tabela "Valabilitate" (ProdusID, DataProducere, DataExpirare).
  + Tabela "Bon" (BonID, CumparatorID, AngajatID, SumaTotala).
  + Tabela "ProduseBon" (BonID, ProdusID, Cantitate).

**Descrierea succintă a tipurilor de asocieri între relații:**

Asocierile dintre tabele sunt realizate prin chei primare și chei străine.

**Asocierea între "Magazin" și "Produse":**

* Tabela "Produse" conține o cheie străină "MagazinID" care se referă la cheia primară "MagazinID" din tabela "Magazin". Această asociere indică că fiecare produs este asociat cu un anumit magazin.

**Asocierea între "Furnizori" și "Produse":**

* Tabela "Produse" conține o cheie străină "FurnizorID" care se referă la cheia primară "FurnizorID" din tabela "Furnizori". Această asociere indică că fiecare produs este furnizat de o anumită companie furnizoare.

**Asocierea între "Angajati" și "Functii":**

* Tabela "Angajati" conține o cheie străină "FunctieID" care se referă la cheia primară "FunctieID" din tabela "Functii". Această asociere indică funcția ocupată de fiecare angajat.

**Asocierea între "Valabilitate" și "Produse":**

* Tabela "Valabilitate" conține o cheie străină "ProdusID" care se referă la cheia primară "ProdusID" din tabela "Produse". Această asociere indică perioada de valabilitate asociată fiecărui produs.

**Asocierea între "Bon" și "Cumparatori":**

* Tabela "Bon" conține o cheie străină "CumparatorID" care se referă la cheia primară "CumparatorID" din tabela "Cumparatori". Această asociere indică cumpărătorul asociat fiecărui bon.

**Asocierea între "Bon" și "Angajati":**

* Tabela "Bon" conține o cheie străină "AngajatID" care se referă la cheia primară "AngajatID" din tabela "Angajati". Această asociere indică angajatul care a procesat fiecare bon.

**Asocierea între "ProduseBon" și "Bon":**

* Tabela "ProduseBon" conține o cheie străină "BonID" care se referă la cheia primară "BonID" din tabela "Bon". Această asociere indică bonul asociat fiecărui produs dintr-un bon.

**Asocierea între "ProduseBon" și "Produse":**

* Tabela "ProduseBon" conține o cheie străină "ProdusID" care se referă la cheia primară "ProdusID" din tabela "Produse". Această asociere indică produsele incluse în fiecare bon, precizând și cantitatea fiecărui produs.

**Anexa B (Script SQL)**

IF EXISTS (SELECT \* FROM sys.databases WHERE name = 'SistemGestionalBon')

BEGIN

USE master;

DROP DATABASE SistemGestionalBon;

END;

CREATE DATABASE SistemGestionalBon;

-- Creare tabela Magazin

CREATE TABLE SistemGestionalBon.dbo.Magazin (

MagazinID TINYINT IDENTITY(1,1) PRIMARY KEY,

NumeMagazin VARCHAR(255),

Adresa VARCHAR(255)

);

-- Inserare date in tabela Magazin

INSERT INTO SistemGestionalBon.dbo.Magazin (NumeMagazin, Adresa)

VALUES

('Mega Store', 'Strada Primaverii, nr. 123'),

('SuperMart', 'Bulevardul Victoriei, nr. 456'),

('Fresh Market', 'Aleea Fructelor, nr. 789'),

('Bonus', 'Bulevardul Unirii, nr. 789'),

('Nr1', 'Strada Luminii, nr. 456'),

('Linella', 'Aleea Viselor, nr. 123'),

('Kaufland', 'Strada Soarelui, nr. 101'),

('Local', 'Bulevardul Lunii, nr. 222'),

('Metro', 'Aleea Stelelor, nr. 333'),

('City Supermarket', 'Strada Norilor, nr. 444'),

('Mall', 'Bulevardul Ploilor, nr. 555'),

('Sun City', 'Aleea Fulgilor, nr. 666'),

('Blue Sky Market', 'Strada Vantului, nr. 777'),

('Nr2', 'Sarmisegetusa, nr.43'),

('Metro 2', 'Dacia, nr21');

-- Creare tabela Furnizori

CREATE TABLE SistemGestionalBon.dbo.Furnizori (

FurnizorID SMALLINT PRIMARY KEY,

Denumire VARCHAR(255),

Adresa VARCHAR(255)

);

-- Inserare date in tabela Furnizori

INSERT INTO SistemGestionalBon.dbo.Furnizori (FurnizorID, Denumire, Adresa)

VALUES

(1, 'Flroare', 'Strada Fagilor, nr. 10'),

(2, 'Carpet', 'Aleea Florilor, nr. 25'),

(3, 'Alimente SRL', 'Bulevardul Rozelor, nr. 40'),

(4, 'Delicios Foods SA', 'Aleea Miresmeilor, nr. 15'),

(5, 'Fresco Distributie Alimentara', 'Strada Perlelor, nr. 30'),

(6, 'BioFarm Produse Organice', 'Bulevardul Colibelor, nr. 45'),

(7, 'Gusturi Traditionale Import-Export', 'Aleea Bijuteriilor, nr. 60'),

(8, 'Marele Pantry Importer', 'Strada Elegantei, nr. 75'),

(9, 'ProNatura Distributie', 'Bulevardul Refined, nr. 90'),

(10, 'Alimex Companie Alimentara', 'Aleea Preciosilor, nr. 105'),

(11, 'EcoBio Distributie Bio', 'Strada Eterna, nr. 120'),

(12, 'MegaFoods Importer', 'Bulevardul Modernitatii, nr. 135'),

(13, 'TopGusturi Distributie', 'Aleea Impecabililor, nr. 150'),

(14, 'ProdusLocal SA', 'Dacia, nr32'),

(15, 'ImportExport', 'Independentei, nr 322');

-- Creare tabela Cumparatori

CREATE TABLE SistemGestionalBon.dbo.Cumparatori (

CumparatorID INT PRIMARY KEY,

Nume VARCHAR(255),

Prenume VARCHAR(255),

Telefon VARCHAR(15)

);

-- Inserare date in tabela Cumparatori

INSERT INTO SistemGestionalBon.dbo.Cumparatori (CumparatorID, Nume, Prenume, Telefon)

VALUES

(1, 'Ion', 'Popescu', '062048221'),

(2, 'Ana', 'Ionescu', '079342896'),

(3, 'Mihai', 'Georgescu', '062097521'),

(4, 'Elena', 'Dumitru', '062048982'),

(5, 'Adrian', 'Popa', '062918221'),

(6, 'Cristina', 'Vasilescu', '069248221'),

(7, 'George', 'Stancu', '062872121'),

(8, 'Andreea', 'Munteanu', '062049989'),

(9, 'Alex', 'Constantinescu', '069898221'),

(10, 'Laura', 'Moldovan', '062098321'),

(11, 'Vlad', 'Alexandrescu', '068676221'),

(12, 'Raluca', 'Radu', '062088221'),

(13, 'Ciprian', 'Iancu', '062023211'),

(14, 'Catalin', 'Popa', '062028456'),

(15, 'Ion', 'Mardarenco', '079325323');

-- Creare tabela Functii

CREATE TABLE SistemGestionalBon.dbo.Functii (

FunctieID INT PRIMARY KEY,

Denumire VARCHAR(255),

SefFunctie VARCHAR(255)

);

-- Inserare date in tabela Functii

INSERT INTO SistemGestionalBon.dbo.Functii (FunctieID, Denumire, SefFunctie)

VALUES

(1, 'Manager', 'Marius Popescu'),

(2, 'Vanzator', 'Andreea Ionescu'),

(3, 'Casier', 'George Dumitrescu'),

(4, 'Receptioner', 'Marius Popescu'),

(5, 'Vanzator secundar', 'Mihai Patic'),

(6, 'Casier principal', 'George Petrache'),

(7, 'Logistician', 'Marius Popescu'),

(8, 'Vanzator principal', 'Cristina Ioana'),

(9, 'Hamal', 'George Ionescu'),

(10, 'Supraveghetor', 'Marius Popescu'),

(11, 'Consultant', 'Catalin Popa'),

(12, 'Sofer', 'George Patic'),

(13, 'Paznic', 'Alexandru Chirica');

-- Creare tabela Angajati

CREATE TABLE SistemGestionalBon.dbo.Angajati (

AngajatID SMALLINT PRIMARY KEY,

Nume VARCHAR(255),

Prenume VARCHAR(255),

AnulNasterii INT,

FunctieID INT,

NumeComplet AS (Nume + ' ' + Prenume) PERSISTED

FOREIGN KEY (FunctieID) REFERENCES Functii(FunctieID)

);

-- Inserare date in tabela Angajati

INSERT INTO SistemGestionalBon.dbo.Angajati (AngajatID, Nume, Prenume, AnulNasterii, FunctieID)

VALUES

(1, 'Popescu', 'Ion', 1980, 1),

(2, 'Negura', 'Ana', 1990, 2),

(3, 'Dumitru', 'Mihai', 1985, 3),

(4, 'Gabriela', 'Vasile', 1982, 4),

(5, 'Radu', 'Elena', 1995, 5),

(6, 'Ana', 'Adrian', 1988, 6),

(7, 'Vasile', 'Cristina', 1980, 7),

(8, 'Cristian', 'Mihai', 1992, 8),

(9, 'Mihai', 'Alexandru', 1985, 9),

(10, 'Adrian', 'Laura', 1987, 10),

(11, 'Elena', 'Vlad', 1990, 11),

(12, 'Iuliana', 'Raul', 1983, 12),

(13, 'Valentin', 'Ciprian', 1986, 13),

(14, 'Popa', 'Galina', 1979, 11),

(15, 'Grigorenco', 'Ion', 1899, 13);

-- Creare tabela Produse

CREATE TABLE SistemGestionalBon.dbo.Produse (

ProdusID INT PRIMARY KEY,

NumeProdus VARCHAR(255),

MagazinID TINYINT,

PretProdus DECIMAL(10, 2),

FurnizorID SMALLINT,

FOREIGN KEY (MagazinID) REFERENCES Magazin(MagazinID),

FOREIGN KEY (FurnizorID) REFERENCES Furnizori(FurnizorID)

);

-- Inserare date in tabela Produse

INSERT INTO SistemGestionalBon.dbo.Produse (ProdusID, NumeProdus, MagazinID, PretProdus, FurnizorID)

VALUES

(1, 'Făină integrală bio', 1, 50.00, 1),

(2, 'Ovăz ecologic', 4, 30.00, 2),

(3, 'Miere de albine pură', 3, 25.00, 3),

(4, 'Quinoa organica', 7, 40.00, 4),

(5, 'Ulei de măsline extravirgin', 5, 28.00, 5),

(6, 'Nuci organice', 8, 22.00, 6),

(7, 'Ciocolată neagră bio', 9, 60.00, 7),

(8, 'Fructe uscate mix', 9, 35.00, 8),

(9, 'Ceai verde organic', 9, 18.00, 9),

(10, 'Mazăre congelată', 10, 45.00, 10),

(11, 'Oțet balsamic premium', 11, 32.00, 11),

(12, 'Lapte de migdale', 12, 25.00, 12),

(13, 'Sos de roșii organic', 8, 55.00, 13),

(14, 'Peste rosu', 13, 52, 4),

(15, 'Apa minerala', 4, 12, 2);

--Creare tabela valabilitate

CREATE TABLE SistemGestionalBon.dbo.Valabilitate (

ProdusID INT PRIMARY KEY,

DataProducere DATE,

DataExpirare DATE,

FOREIGN KEY (ProdusID) REFERENCES Produse(ProdusID)

);

-- Inserare date in tabela Valabilitate

INSERT INTO SistemGestionalBon.dbo.Valabilitate (ProdusID, DataProducere, DataExpirare)

VALUES

(1, '2022-01-01', '2022-12-31'),

(2, '2021-05-15', '2022-05-15'),

(3, '2022-03-10', '2023-03-10'),

(4, '2021-11-15', '2022-11-15'),

(5, '2022-06-20', '2023-06-20'),

(6, '2021-08-05', '2022-08-05'),

(7, '2022-02-28', '2023-02-28'),

(8, '2021-12-10', '2022-12-10'),

(9, '2022-04-25', '2023-04-25'),

(10, '2022-07-01', '2023-07-01'),

(11, '2021-09-12', '2022-09-12'),

(12, '2022-01-05', '2023-01-05'),

(13, '2022-10-30', '2023-10-30'),

(14, '2022-11-30', '2024-10-30'),

(15, '2023-12-02', '2024-10-30');

-- Creare tabela Bon

CREATE TABLE SistemGestionalBon.dbo.Bon (

BonID INT PRIMARY KEY,

CumparatorID INT,

AngajatID SMALLINT,

SumaTotala DECIMAL(10, 2),

FOREIGN KEY (CumparatorID) REFERENCES Cumparatori(CumparatorID),

FOREIGN KEY (AngajatID) REFERENCES Angajati(AngajatID)

);

-- Inserare date in tabela Bon

INSERT INTO SistemGestionalBon.dbo.Bon (BonID, CumparatorID, AngajatID, SumaTotala)

VALUES

(1, 1, 1, 110.00),

(2, 2, 2, 75.00),

(3, 3, 13, 50.00),

(4, 2, 2, 130.00),

(5, 5, 3, 90.00),

(6, 13, 10, 115.00),

(7, 8, 2, 78.00),

(8, 9, 9, 55.00),

(9, 8, 1, 75.00),

(10, 6, 2, 105.00),

(11, 3, 2, 68.00),

(12, 13, 2, 40.00),

(13, 12, 6, 50.00),

(14, 15, 5, 12),

(15, 3, 1, 100);

-- Creare tabela ProduseBon

CREATE TABLE SistemGestionalBon.dbo.ProduseBon (

BonID INT,

ProdusID INT,

Cantitate INT,

PRIMARY KEY (BonID, ProdusID),

FOREIGN KEY (BonID) REFERENCES Bon(BonID),

FOREIGN KEY (ProdusID) REFERENCES Produse(ProdusID)

);

-- Inserare date in tabela ProduseBon

INSERT INTO SistemGestionalBon.dbo.ProduseBon (BonID, ProdusID, Cantitate)

VALUES

(1, 1, 2),

(1, 2, 1),

(2, 3, 3),

(2, 2, 2),

(3, 1, 1),

(3, 3, 2),

(4, 4, 3),

(4, 5, 1),

(5, 6, 2),

(5, 7, 1),

(6, 8, 3),

(6, 9, 1),

(7, 10, 2);

--va afisa produsul si denumirea furnizorului

SELECT a.NumeProdus, b.Denumire as furnizor

FROM SistemGestionalBon.dbo.Produse a

INNER JOIN SistemGestionalBon.dbo.Furnizori b ON a.FurnizorID = b.FurnizorID;

--va afisa cumaraotrii si suma totala a bonuriloe

SELECT Cumparatori.Nume, Cumparatori.Prenume, Bon.SumaTotala

FROM SistemGestionalBon.dbo.Cumparatori

LEFT JOIN SistemGestionalBon.dbo.Bon ON Cumparatori.CumparatorID = Bon.CumparatorID;

--Afișează denumirea funcțiilor și numele complet al angajaților. Funcțiile fără angajați vor apărea și ele în rezultat.

SELECT Functii.Denumire, Angajati.NumeComplet

FROM SistemGestionalBon.dbo.Functii

RIGHT JOIN SistemGestionalBon.dbo.Angajati ON Functii.FunctieID = Angajati.FunctieID;

--Afișează numele magazinelor, numele produselor și prețurile produselor pentru acele produse care au un preț mai mare de 30.00.

SELECT Magazin.NumeMagazin, Produse.NumeProdus, Produse.PretProdus

FROM SistemGestionalBon.dbo.Magazin

INNER JOIN SistemGestionalBon.dbo.Produse ON Magazin.MagazinID = Produse.MagazinID

WHERE Produse.PretProdus > 30.00;

--Afișează numele complet al angajaților, denumirea funcțiilor și suma totală a bonurilor asociate.

SELECT Angajati.NumeComplet, Functii.Denumire, sum(Bon.SumaTotala)

FROM SistemGestionalBon.dbo.Angajati

INNER JOIN SistemGestionalBon.dbo.Functii ON Angajati.FunctieID = Functii.FunctieID

INNER JOIN SistemGestionalBon.dbo.Bon ON Angajati.AngajatID = Bon.AngajatID

group by Angajati.NumeComplet, Functii.Denumire;

--Afișează numele și prenumele cumpărătorilor care au un număr de telefon care începe cu '062'.

SELECT Nume, Prenume, Telefon

FROM SistemGestionalBon.dbo.Cumparatori

WHERE Telefon LIKE '062%';

--Afișează media duratei de valabilitate (în zile) a produselor din tabela de valabilitate.

SELECT AVG(DATEDIFF(day, DataProducere, DataExpirare)) AS MedieDurataValabilitate

FROM SistemGestionalBon.dbo.Valabilitate;

--Afișează denumirile și adresele furnizorilor și magazinelor într-un singur set distinct de rezultate.

SELECT Denumire, Adresa FROM SistemGestionalBon.dbo.Furnizori

UNION

SELECT NumeMagazin, Adresa FROM SistemGestionalBon.dbo.Magazin;

--Afișează numele complet al angajaților care au asociate bonuri în tabela Bon.

SELECT NumeComplet

FROM SistemGestionalBon.dbo.Angajati AS A

WHERE EXISTS (

SELECT 1

FROM SistemGestionalBon.dbo.Bon AS B

WHERE A.AngajatID = B.AngajatID

);

--Afișează numărul de produse pentru fiecare bon, folosind funcția COUNT și GROUP BY.

SELECT a.BonID, COUNT(a.ProdusID) AS NumarProduse

FROM SistemGestionalBon.dbo.ProduseBon a

GROUP BY a.BonID;

--Afișează numele și prenumele cumpărătorilor care nu au asociate bonuri.

SELECT Cumparatori.Nume, Cumparatori.Prenume

FROM SistemGestionalBon.dbo.Cumparatori

LEFT JOIN SistemGestionalBon.dbo.Bon ON Cumparatori.CumparatorID = Bon.CumparatorID

WHERE Bon.BonID IS NULL;

--Afișează produsul și data maximă de expirare asociată fiecărui produs.

SELECT ProdusID, MAX(DataExpirare) AS DataExpirareMaxima

FROM SistemGestionalBon.dbo.Valabilitate

GROUP BY ProdusID;

--Afișează numărul total de produse cumpărate de fiecare cumpărător.

SELECT B.CumparatorID, COUNT(PB.ProdusID) AS NumarProduse

FROM SistemGestionalBon.dbo.Bon AS B

INNER JOIN SistemGestionalBon.dbo.ProduseBon AS PB ON B.BonID = PB.BonID

GROUP BY B.CumparatorID;

--Afișează numele și prețul produselor care au un preț mai mare decât media prețurilor produselor.

SELECT NumeProdus, PretProdus

FROM SistemGestionalBon.dbo.Produse

WHERE PretProdus > (SELECT AVG(PretProdus) FROM SistemGestionalBon.dbo.Produse);

--Afișează ID-ul cumpărătorului și suma totală a cumpărăturilor pentru cei cu cumpărături mai mari de 80.00

SELECT B.CumparatorID, SUM(P.PretProdus) AS SumaTotalaCumparaturi

FROM SistemGestionalBon.dbo.Bon AS B

INNER JOIN SistemGestionalBon.dbo.ProduseBon AS PB ON B.BonID = PB.BonID

INNER JOIN SistemGestionalBon.dbo.Produse AS P ON PB.ProdusID = P.ProdusID

GROUP BY B.CumparatorID

HAVING SUM(P.PretProdus) > 80.00;

--Afișează numele complet al angajaților și vârsta lor actuală, calculată pe baza anului de naștere.

SELECT NumeComplet, DATEPART(YEAR, GETDATE()) - AnulNasterii AS Varsta

FROM SistemGestionalBon.dbo.Angajati

where DATEPART(YEAR, GETDATE()) - AnulNasterii < 50;

--Afișează numele complet al angajaților folosind funcția CONCAT.

SELECT CONCAT(Nume, ' ', Prenume) AS NumeComplet

FROM SistemGestionalBon.dbo.Angajati;

--Afișează numele și prenumele cumpărătorilor care au asociate bonuri.

SELECT Nume, Prenume

FROM SistemGestionalBon.dbo.Cumparatori

WHERE CumparatorID IN (SELECT CumparatorID FROM SistemGestionalBon.dbo.Bon);

--Afișează produsele și perioada de valabilitate care se află între anumite date de producere.

SELECT ProdusID, DataProducere, DataExpirare

FROM SistemGestionalBon.dbo.Valabilitate

WHERE DataProducere BETWEEN '2022-01-01' AND '2022-12-31';