PROIECT

BAZE DE DATE

Microsoft SQL Server

Student: David George-Bogdan

Informatica, Anul I

Universitatea din Pitesti, Facultatea de Stiinte, Educatie Fizica si Informatica

Elaborarea modelului relational al datelor

Se doreste realizarea unui sistem informatic pentru gestionarea comenzilor si a produselor unui magazin online.

Se considera urmatoarea colectie de date:

codClient, numeClient, prenumeClient, telefonClient, emailClient, adresaClient, codProdus, numeProdus, descriereProdus, pretProdus, stockProdus, nrFactura, dataFactura, totalFactura, discountFactura codCategorie, numeCategorie, descriereCategorie, codDistribuitor, numeDistribuitor.

Reguli de gestiune:

1. O factura poate contine mai multe produse, un produs poate fi pe mai multe facturi.
2. Un client poate avea mai multe facturi, o factura poate fi emisa catre un singur client.
3. Un distribuitor poate distribui mai mult produse, un produs poate fi distribuit de un distribuitor.
4. O categorie poate contine mai multe produse, un produs poate face parte dintr-o singura categorie.

Sa se elaboreze modelul relational al datelor:

1. Identificarea entitatilor:
   1. Client
   2. Produs
   3. Factura
   4. Categorie
   5. Distribuitor
2. Identificarea atributelor si a identificatorilor entitatilor:
   1. Client (codClient, numeClient, prenumeClient, telefonClient, emailClient, adresaClient)
   2. Produs (codProdus, numeProdus, descriereProdus, pretProdus, stockProdus)
   3. Factura (nrFactura, dataFactura, totalFactura, discountFactura)
   4. Categorie (codCategorie, numeCategorie, descriereCategorie)
   5. Distribuitor (codDistribuitor, numeDistribuitor)

3. Identificarea asocierilor si a cardinalitatilor lor:

a. Intre Factura si Produs exista asocierea Contine de cardinalitate maxima n:n si de cardinalitate minima 1:1. Asocierea Contine este caracterizata prin atributul specific cantitateVanduta.

b. Intre Client si Factura exista asocierea Primeste de cardinalitate maxima 1:n si de cardinalitate minima 1:1.

c. Intre Distribuitor si Produs exista asocierea Distribuie de cardinalitate maxima 1:n si de cardinalitate minima 1:1.

d. Intre Categorie si Produs exista asocierea Contine2 de cardinalitate maxima 1:n si de cardinalitate minima 1:1.

B. Modelul relational

1. Entitatile devin tabele:

a. tClient (codClient, numeClient, prenumeClient, telefonClient, emailClient, adresaClient)

b. tProdus (codProdus, numeProdus, descriereProdus, pretProdus, stockProdus)

c. tFactura (nrFactura, dataFactura, totalFactura, discountFactura)

d. tCategorie (codCategorie, numeCategorie, descriereCategorie)

e. tDistribuitor (codDistribuitor, numeDistribuitor)

2. Rezolvarea asocierilor:

1. 1. Asocierea (Contine) de cardinalitate maxima n:n se rezolva prin intermediul unui tabel asociativ. Se preiau cheile primare ale tabelelor asociate care devin chei straine in noul tabel, se adauga atributele specifice asocierii si se stabileste cheia primara a noului tabel pe baza celor dour chei straine si a altor atribute.

tDetaliiFactura (idVanzare, #nrFactura, #codProdus, cantitateVanduta)

2. Asocierea (Primeste) de cardinalitate 1:n se rezolva cu ajutorul cheii straine Cheia primara din tabelul (Client) devine cheie straina in tabelul (Factura)

tFactura (nrFactura, dataFactura, #codClient, totalFactura, discountFactura)

3. Asocierea (Distribuie) de cardinalitate 1:n se rezolva cu ajutorul cheii straine. Cheia primara din (Distribuitor) devine cheie straina in tabelul (Produs)

tProdus (codProdus, numeProdus, descriereProdus, pretProdus, stockProdus, #codDistribuitor)

4. Asocierea (Contine2) de cardinalitate maxima 1:n se rezolva cu ajutorul cheii straine. Cheia primara din tabelul (Categorii) devine cheie straina in tabelul (Produs)

tProdus (codProdus, numeProdus, descriereProdus, pretProdus, stockProdus, #codDistribuitor, #codCategorie)

Finalizare:

a. tClient (codClient, numeClient, prenumeClient, telefonClient, emailClient, adresaClient)

b. tProdus (codProdus, numeProdus, descriereProdus, pretProdus, stockProdus, #codDistribuitor, #codCategorie)

c. tFactura (nrFactura, dataFactura, #codClient, totalFactura, discountFactura)

d. tCategorie (codCategorie, numeCategorie, descriereCategorie)

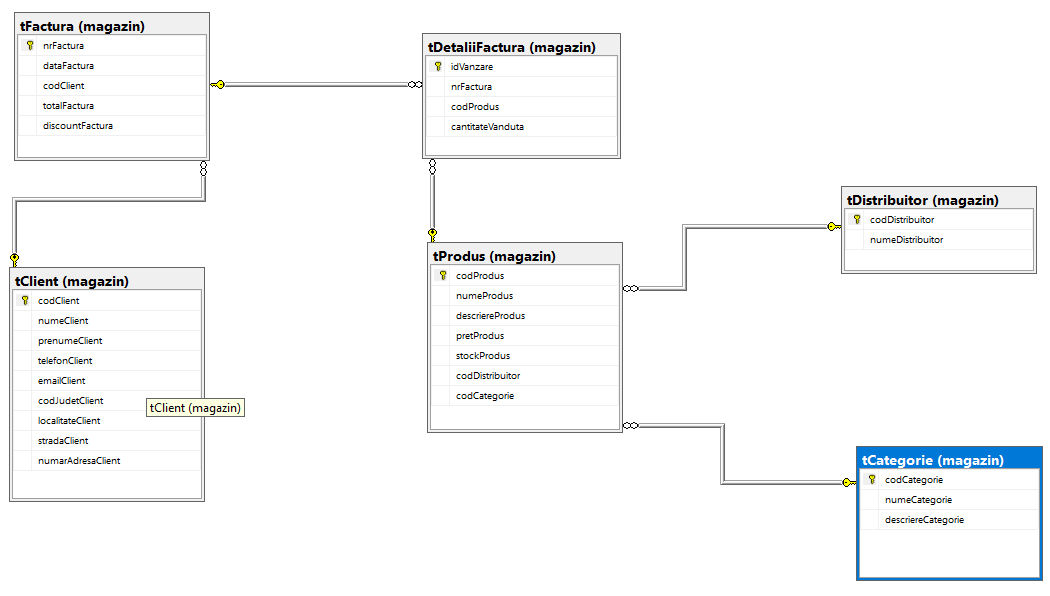
e. tDistrbuitor (codDistibuitor, numeDistribuitor)

f. tDetaliiFactura (idVanzare, #nrFactura, #codProdus, cantitateVanduta)

Normalizarea relatiilor:

tClient nu este in FN1, deoarece atributul “Adresa” este compus. Vom descompune atributul in componentele sale atomice: codJudetClient, localitateClient, stradaClient, numarAdresaClient.

tClient (codClient, numeClient, prenumeClient, telefonClient, emailClient, codJudetClient, localitateClient, stradaClient, numarAdresaClient)



Crearea tabelelor

--Crearea bazei de date dbMagazinOnline

create database dbMagazinOnline

--Activarea bazei de date dbMagazinOnline

use dbMagazinOnline

--Crearea schemei magazin

create schema magazin

--Crearea tabelului tClient in interiorul schemei magazin

create table magazin.tClient

(

codClient char(10) not null primary key,

numeClient varchar(60) not null,

prenumeClient varchar(60) not null,

telefonClient char(15) not null,

emailClient varchar(60),

codJudetClient char(2) not null,

localitateClient varchar(60) not null,

stradaClient varchar(60) not null,

numarAdresaClient char(5)

)

--Crearea tabelului tFactura in interiorul schemei magazin

create table magazin.tFactura

(

nrFactura char(10) not null primary key,

dataFactura date,

codClient char (10) not null constraint fk\_Client references magazin.tClient(codClient),

totalFactura decimal,

discountFactura int

)

--Crearea tabelului tDistribuitor in interiorul schemei magazin

create table magazin.tDistribuitor

(

codDistribuitor char(10) not null primary key,

numeDistribuitor varchar(60) not null

)

--Crearea tabelului tCategorie in interiorul schemei magazin

create table magazin.tCategorie

(

codCategorie char(10) not null primary key,

numeCategorie varchar(60) not null,

descriereCategorie varchar(500)

)

--Crearea tabelului tProdus in interiorul schemei magazin

create table magazin.tProdus

(

codProdus char(10) not null primary key,

numeProdus varchar(60) not null,

descriereProdus varchar(500),

pretProdus decimal,

stockProdus int,

codDistribuitor char(10) not null constraint fk\_Distribuitor references magazin.tDistribuitor(codDistribuitor),

codCategorie char(10) not null constraint fk\_Categorie references magazin.tCategorie(codCategorie)

)

--Crearea tabelului magazin.tDetaliiFactura

create table magazin.tDetaliiFactura

(

idVanzare char(10),

nrFactura char(10) not null constraint fk\_Factura references magazin.tFactura(nrFactura),

codProdus char(10) not null constraint fk\_ProdusDetaliiFactura references magazin.tProdus(codProdus),

cantitateVanduta int

)

Modificarea structurii tabelelor

--Eliminati coloana Localitate din tabelul tClienti

alter table magazin.tClient

drop column Localitate

--Adaugati tabelului tProdus coloana OutOfStock + constrangera check('DA', 'NU')

alter table magazin.tProdus

add OutOfStock varchar(10) check (OutOfStock = 'DA' or 'NU'

--Sa se modifice tipul coloanei totalFactura din decimal in float, afltat in magazin.tFactura

alter table magazin.tFactura

alter column totalFactura float

--Utilizand procedura stocata de sistem sp\_rename sa se redenumeasca coloana 'emailClient' din magazin.tClienti in 'adresaEmailClient'

execute sp\_rename 'magazin.tClient.emailClient', 'adresaEmailClient', 'column'

--Adaugati constrangerea not null + constrangerea primary key in coloana idVanzare din tabelul magazin.tDetaliiFactura

alter table magazin.tDetaliiFactura

alter column idVanzare char(10) not null constraint pk\_idVanzare primary key

Folosirea comenzii “INSERT”

--Inserarea datelor in tabelul magazin.tClient

insert into magazin.tClient values ('C001', 'David', 'Bogdan', '+40743030002', 'davidgeorgebogdan@gmail.com', 'AG', 'Pitesti', 'Trivale', '90')

insert into magazin.tClient values ('C002', 'Popescu', 'Ion', '+40745254638', 'popescuion@gmail.com', 'AG', 'Pitesti', 'Lalelelor', '25')

insert into magazin.tClient values ('C003', 'Ionescu', 'Alina', '+40746564334', 'ionescualina@gmail.com', 'DB', 'Targoviste', 'Bulevardul Unirii', '45')

insert into magazin.tClient values ('C004', 'Mihalache', 'Ionela', '+40746535665', 'mihalacheionela@gmail.com', 'CJ', 'Cluj', 'Ialomita', '52')

insert into magazin.tClient values ('C005', 'Popa', 'Andrei', '+40748756394', 'popaandrei@gmail.com', 'DJ', 'Craiova', '3 Februarie', 'A78')

insert into magazin.tClient values ('C006', 'Nitu', 'Adrian', '+40741234535', 'nituadrian@gmail.com', 'BV', 'Brasov', 'Albinelor', '23')

insert into magazin.tClient values ('C007', 'Muresan', 'Ioana', '+40789467364', 'muresanioana@gmail.com', 'SB', 'Sibiu', 'Aleea Buia', NULL)

insert into magazin.tClient values ('C008', 'Nistor', 'Mihai', '+40735685673', NULL , 'AG', 'Pitesti', 'Strada Basarabia', '67')

insert into magazin.tClient values ('C010', 'Popescu', 'George', '+40737598142', 'popescugeorge@gmail.com', 'TM', 'Timisoara', 'Aleea Inului', '52')

insert into magazin.tClient values ('C011', 'Sava', 'Maria', '+40723465698', 'savamaria@gmail.com', 'B', 'Bucuresti', 'Dorin Pavel', 'B15')

insert into magazin.tClient values ('C012', 'Negoita', 'Stefan', '+40763845973', 'negoitastefan@gmail.com', 'IS', 'Iasi', 'Milcov', NULL)

insert into magazin.tClient values ('C013', 'Dinescu', 'Georgiana', '+40765298734', 'dinescugeorgiana@gmail.com', 'CJ', 'Cluj', 'Cernei', '75')

insert into magazin.tClient values ('C014', 'Iancu', 'Marian', '+40756234549', NULL, 'CT', 'Constanta', 'Aleea Afinei', NULL)

insert into magazin.tClient values ('C015', 'Toma', 'Stefan', '+40723464356', 'tomastefan@gmail.com', 'AG', 'Pitesti', 'Dragos Voda', '23')

insert into magazin.tClient values ('C016', 'Predescu', 'Paul', '+40743543239', 'predescupaul@gmail.com', 'AG', 'Pitesti', 'Eremia Grigorescu', '26')

insert into magazin.tClient values ('C017', 'Ghita', 'Andrei', '+40748735458', 'ghitaandrei@gmail.com', 'CJ', 'Cluj', 'Izvorului', '97')

insert into magazin.tClient values ('C018', 'Mocanu', 'Alexandra', '+40743986274', 'mocanualexandra@gmail.com', 'CT', 'Constanta', 'Frunzelor', NULL)

insert into magazin.tClient values ('C019', 'Popescu', 'Teodora', '+40743235487', NULL, 'TM', 'Timisoara', 'Iris', '16')

insert into magazin.tClient values ('C020', 'Albu', 'Laurentiu', '+40735567928', 'albulaurentiu@gmail.com', 'AG', 'Pitesti', 'Gheorghe Sincai', NULL)

insert into magazin.tClient values ('C021', 'Marin', 'Gabriel', '+40747987264', 'maringabriel@gmail.com', 'SB', 'Sibiu', 'Cucului', '92')

insert into magazin.tClient values ('C022', 'Stoica', 'Vlad', '+40745345328', 'stoicavlad@gmail.com', 'B', 'Bucuresti', 'Dr. I. Cantacuzino', '48')

insert into magazin.tClient values ('C023', 'Moisescu', 'Larisa', '+40752790912', NULL, 'BV', 'Brasov', 'Alexandru cel Bun', NULL)

insert into magazin.tClient values ('C024', 'Manole', 'Alexandru', '+40745247945', 'manolealexandru@gmail.com', 'AG', 'Pitesti', 'Pandele', '13')

insert into magazin.tClient values ('C025', 'Tomescu', 'Catalina', '+40734982374', 'tomescucatalina@gmail.com', 'SB', 'Sibiu', 'Ioan Ratiu', '52')

insert into magazin.tClient values ('C026', 'Dobre', 'Ioana', '+40738728931', 'dobreioana@gmail.com', 'CT', 'Constanta', 'Baladei', '27')

insert into magazin.tClient values ('C027', 'Constantinescu', 'Ionel', '+40782563984', 'constantinescuionel@gmail.com', 'TM', 'Timisoara', 'Gheorghe Ranetti', '50')

insert into magazin.tClient values ('C028', 'Mihalache', 'Mirela', '+40734597185', 'mihalachemirela@gmail.com', 'DJ', 'Craiova', 'Calea Dunarii', '25')

insert into magazin.tClient values ('C029', 'Iancu', 'Andrei', '+40789673895', 'iancuandrei@gmail.com', 'B', 'Bucuresti', 'Carol Davila', '61')

insert into magazin.tClient values ('C030', 'Vladimirescu', 'Tudor', '+40789263918', 'vladimirescutudor@gmail.com', 'AG', 'Pitesti', 'George Sion', '27')

insert into magazin.tClient values ('C031', 'Tabacu', 'Vlad', '+40749784783', NULL, 'CJ', 'Cluj', 'Dem Radulescu', '62')

--Inserarea datelor in tabelul magazin.tFactura

insert into magazin.tFactura values ('F001', '2022-01-02', 'C025', 200.98, NULL)

insert into magazin.tFactura values ('F002', '2022-01-04', 'C018', 873.29, NULL)

insert into magazin.tFactura values ('F003', '2022-01-05', 'C023', 765.18, NULL)

insert into magazin.tFactura values ('F004', '2022-01-06', 'C004', 678.72, NULL)

insert into magazin.tFactura values ('F005', '2022-01-10', 'C023', 932.10, NULL)

insert into magazin.tFactura values ('F006', '2022-01-21', 'C013', 200.81, NULL)

insert into magazin.tFactura values ('F007', '2022-02-10', 'C026', 98.29, NULL)

insert into magazin.tFactura values ('F008', '2022-02-15', 'C004', 735.29, NULL)

insert into magazin.tFactura values ('F009', '2022-02-16', 'C026', 1418.92, 10)

insert into magazin.tFactura values ('F010', '2022-02-19', 'C027', 1169.37, 10)

insert into magazin.tFactura values ('F011', '2022-02-25', 'C012', 336.81, NULL)

insert into magazin.tFactura values ('F012', '2022-02-27', 'C005', 87.20, NULL)

insert into magazin.tFactura values ('F013', '2022-03-04', 'C003', 289.58, NULL)

insert into magazin.tFactura values ('F014', '2022-03-05', 'C015', 4466.27, 20)

insert into magazin.tFactura values ('F015', '2022-03-10', 'C002', 574.10, NULL)

insert into magazin.tFactura values ('F016', '2022-03-07', 'C017', 180.20, NULL)

insert into magazin.tFactura values ('F017', '2022-03-09', 'C007', 989.64, NULL)

insert into magazin.tFactura values ('F018', '2022-03-31', 'C014', 401.19, NULL)

insert into magazin.tFactura values ('F019', '2022-04-01', 'C027', 55.82, NULL)

insert into magazin.tFactura values ('F020', '2022-04-02', 'C029', 689.23, NULL)

insert into magazin.tFactura values ('F021', '2022-04-03', 'C017', 354.15, NULL)

insert into magazin.tFactura values ('F022', '2022-04-06', 'C023', 160.92, NULL)

insert into magazin.tFactura values ('F023', '2022-04-08', 'C020', 131.64, NULL)

insert into magazin.tFactura values ('F024', '2022-04-09', 'C028', 434.32, NULL)

insert into magazin.tFactura values ('F025', '2022-04-10', 'C014', 12000, 15)

insert into magazin.tFactura values ('F026', '2022-04-11', 'C001', 281, NULL)

insert into magazin.tFactura values ('F027', '2022-04-11', 'C024', 1219, 15)

insert into magazin.tFactura values ('F028', '2022-04-20', 'C030', 803, NULL)

insert into magazin.tFactura values ('F029', '2022-04-20', 'C002', 613, NULL)

insert into magazin.tFactura values ('F030', '2022-04-20', 'C019', 2943, 15)

--Inserarea datelor in tabelul magazin.tCategorie

insert into magazin.tCategorie values ('Cat001', 'Laptop', NULL)

insert into magazin.tCategorie values ('Cat002', 'Componente', 'Placi video, Procesoare, Placi de baza, SSD-uri, Surse')

insert into magazin.tCategorie values ('Cat003', 'Periferice', 'Mouse-uri, Tastaturi, Boxe, Casti, Camere web')

insert into magazin.tCategorie values ('Cat004', 'Software', NULL)

insert into magazin.tCategorie values ('Cat005', 'Retelistica', 'Routere, Nas-uri')

--Inserara datelor in tabelul magazin.tDistribuitor

insert into magazin.tDistribuitor

values ('D001', 'DGB SRL'), ('D002', 'QPX Logistics SRL'), ('D003', 'ABC Shipping SRL'), ('D004', 'Shot Tech SRL'), ('D005', 'ChipTech Company SRL')

--Inserarea datelor in tabelul magazin.tProdus

insert into magazin.tProdus values ('P001', 'Apple MacBook Pro 14', NULL, 12000.89, 75, 'D001', 'Cat001' )

insert into magazin.tProdus values ('P002', 'ASUS TUF A15', 'AMD Ryzen™ 5 4600H, 15.6", Full HD', 5000.45 , 80, 'D001', 'Cat001')

insert into magazin.tProdus values ('P003', 'ASUS X515EA', 'Intel® Core™ i5-1135G7', 2434.84, 20, 'D001','Cat001')

insert into magazin.tProdus values ('P004', 'Gigabyte GeForce® RTX™ 3080', '10GB GDDR6X, 320-bit', 6323.89 , 5, 'D002', 'Cat002')

insert into magazin.tProdus values ('P005', 'AMD Ryzen™ 7 5800X', '36MB, 4.7GHz, Socket AM4', 1736.86, 45, 'D002', 'Cat002')

insert into magazin.tProdus values ('P006', 'Asus ROG STRIX B550-F GAMING', 'Socket AM4', 1164.32, 28, 'D002', 'Cat002')

insert into magazin.tProdus values ('P007', 'ASUS PRIME B550M-K', 'Socket AM4', 602.15, 100, 'D002', 'Cat002')

insert into magazin.tProdus values ('P008', 'Asus ROG STRIX B550-A GAMING', 'Socket AM4', 1021.62, 64, 'D002', 'Cat002')

insert into magazin.tProdus values ('P010', 'Kingston A400', 'Solid State Drive (SSD) 480GB, 2.5"', 250, 190, 'D002', 'Cat002')

insert into magazin.tProdus values ('P011', 'Samsung 970 EVO Plus', 'Solid state drive (SSD) 1TB, NVMe, M.2.', 591.99, 72, 'D002', 'Cat002')

insert into magazin.tProdus values ('P012', 'ADATA SU630', 'Solid-State Drive (SSD) 240GB, 2.5", SATA III', 142.72, 150, 'D002', 'Cat002')

insert into magazin.tProdus values ('P013', 'Seasonic 750W', 'OCUS GX Series, 80 PLUS Gold, Full Modulara, ATX v2.4', 605.10, 26, 'D002', 'Cat002')

insert into magazin.tProdus values ('P014', 'Woden 750', '750W Real Power, PFC Activ, 80 Plus Gold', 318.99, 91, 'D002', 'Cat002')

insert into magazin.tProdus values ('P015', 'Razer DeathAdder', 'Mouse', 90, 82, 'D003', 'Cat003')

insert into magazin.tProdus values ('P016', 'Logitech G102 Lightsync', '8000 dpi, RGB, Negru', 109.99, 14, 'D003', 'Cat003')

insert into magazin.tProdus values ('P017', 'ASUS WT425', '1600 dpi, USB, Alb', 48.99, 3, 'D003', 'Cat003')

insert into magazin.tProdus values ('P018', 'Logitech Pebble M350', 'Mouse wireless', 94.99, 23, 'D003', 'Cat003')

insert into magazin.tProdus values ('P019', 'Redragon Surara', NULL, 169.99, 42, 'D003', 'Cat003')

insert into magazin.tProdus values ('P020', 'HyperX Alloy Origins', 'Tastatura gaming mecanica', 264.99, 62, 'D003', 'Cat003')

insert into magazin.tProdus values ('P021', 'Logitech Multimedia Speakers', 'Z333, 40W RMS, Negru', 299.99, 38, 'D003', 'Cat003')

insert into magazin.tProdus values ('P022', 'Edifier 2.0 256W S3000PRO', 'Boxe bluetooth, USB, telecomanda, brown', 2184.99, 2, 'D003', 'Cat003')

insert into magazin.tProdus values ('P023', 'Edifier 2.0 66W R1700BT', 'Boxe bluetooth, Black', 589.99, 12, 'D003', 'Cat003')

insert into magazin.tProdus values ('P024', 'HyperX Cloud II Pro', 'Casti Gun Metal', 217.99, 1, 'D003', 'Cat003')

insert into magazin.tProdus values ('P025', 'Razer Kraken 2019', 'Casti multiplatforma, Negru', 199.99, 26, 'D003', 'Cat003')

insert into magazin.tProdus values ('P026', 'Razer Hammerhead True Wireless', 'Casti mobile gaming Bluetooth 5.0, Negru', 264.99, 19, 'D003', 'Cat003')

insert into magazin.tProdus values ('P027', 'Logitech G PRO X', 'Casti functie BLUE VO!CE, Negru', 498.99, 23, 'D003', 'Cat003')

insert into magazin.tProdus values ('P028', 'Logitech HD Pro C920', 'Camera web, Full HD, Negru', 369.29, 9, 'D003', 'Cat003')

insert into magazin.tProdus values ('P029', 'BabyToy™ CX-W10 Pro', 'Webcam, Dual Microfon, Auto-Focus', 159.62, 2, 'D003', 'Cat003')

insert into magazin.tProdus values ('P030', 'Windows 11 Pro', NULL, 80, 190, 'D004', 'Cat004')

insert into magazin.tProdus values ('P031', 'Microsoft Office 2019', NULL, 49.50, 254, 'D004', 'Cat004')

insert into magazin.tProdus values ('P032', 'Bitdefender Internet Security 2021', NULL, 79.99, 300, 'D004', 'Cat004')

insert into magazin.tProdus values ('P033', 'TP-Link Archer C80', 'Router wireless Full Gigabit, Dual Band, MU-MIMO, Wi-Fi Wave2', 169.99, 82, 'D005', 'Cat005')

insert into magazin.tProdus values ('P034', 'TP-Link Archer AX20', 'Dual-Band Gigabit AX1800, 1.8 Gbps, cu OFDMA', 319.99, 15, 'D005', 'Cat005')

insert into magazin.tProdus values ('P035', 'Synology DS920+', NULL, 3271, 1, 'D005', 'Cat005')

insert into magazin.tProdus values ('P036', 'QNAP TS-251D-2G', NULL, 1499.99, 7, 'D005', 'Cat005')

--Inserarea datelor in tabelul magazin.tDetaliiFactura

insert into magazin.tDetaliiFactura values ( 'DF001', 'F001', 'P015', 2)

insert into magazin.tDetaliiFactura values ( 'DF002', 'F002', 'P025', 4)

insert into magazin.tDetaliiFactura values ( 'DF003', 'F003', 'P007', 1)

insert into magazin.tDetaliiFactura values ( 'DF004', 'F004', 'P026', 2)

insert into magazin.tDetaliiFactura values ( 'DF005', 'F005', 'P028', 3)

insert into magazin.tDetaliiFactura values ( 'DF006', 'F006', 'P029', 1)

insert into magazin.tDetaliiFactura values ( 'DF007', 'F007', 'P015', 1)

insert into magazin.tDetaliiFactura values ( 'DF008', 'F008', 'P013', 1)

insert into magazin.tDetaliiFactura values ( 'DF009', 'F009', 'P036', 1)

insert into magazin.tDetaliiFactura values ( 'DF010', 'F010', 'P007', 2)

insert into magazin.tDetaliiFactura values ( 'DF011', 'F011', 'P034', 1)

insert into magazin.tDetaliiFactura values ( 'DF012', 'F012', 'P030', 1)

insert into magazin.tDetaliiFactura values ( 'DF013', 'F013', 'P010', 1)

insert into magazin.tDetaliiFactura values ( 'DF014', 'F014', 'P003', 2)

insert into magazin.tDetaliiFactura values ( 'DF015', 'F015', 'P020', 2)

insert into magazin.tDetaliiFactura values ( 'DF016', 'F016', 'P033', 1)

insert into magazin.tDetaliiFactura values ( 'DF017', 'F017', 'P028', 2)

insert into magazin.tDetaliiFactura values ( 'DF018', 'F018', 'P028', 1)

insert into magazin.tDetaliiFactura values ( 'DF019', 'F019', 'P031', 1)

insert into magazin.tDetaliiFactura values ( 'DF020', 'F020', 'P007', 1)

insert into magazin.tDetaliiFactura values ( 'DF021', 'F021', 'P012', 2)

insert into magazin.tDetaliiFactura values ( 'DF022', 'F022', 'P029', 1)

insert into magazin.tDetaliiFactura values ( 'DF023', 'F023', 'P012', 1)

insert into magazin.tDetaliiFactura values ( 'DF024', 'F024', 'P014', 1)

insert into magazin.tDetaliiFactura values ( 'DF025', 'F025', 'P001', 1)

insert into magazin.tDetaliiFactura values ( 'DF026', 'F026', 'P034', 1)

insert into magazin.tDetaliiFactura values ( 'DF027', 'F027', 'P005', 1)

insert into magazin.tDetaliiFactura values ( 'DF028', 'F028', 'P027', 2)

insert into magazin.tDetaliiFactura values ( 'DF029', 'F029', 'P007', 1)

insert into magazin.tDetaliiFactura values ( 'DF030', 'F030', 'P003', 1)

Folosirea comenzii “SELECT”

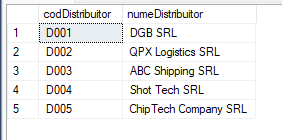
--Selectati numele clientilor, prenumele si codul judetului din tabelul tClient

select numeClient,prenumeClient,codJudetClient from magazin.tClient



--Selectati toate coloanele din tabelul tDistribuitor

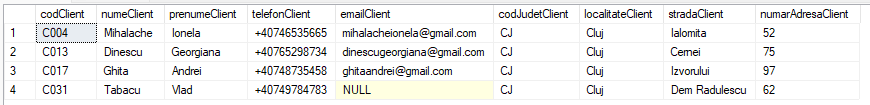
select \* from magazin.tDistribuitor



--Selectati toate coloanele din tabelul tClient pentru clientii din judetul Cluj

select \* from magazin.tClient

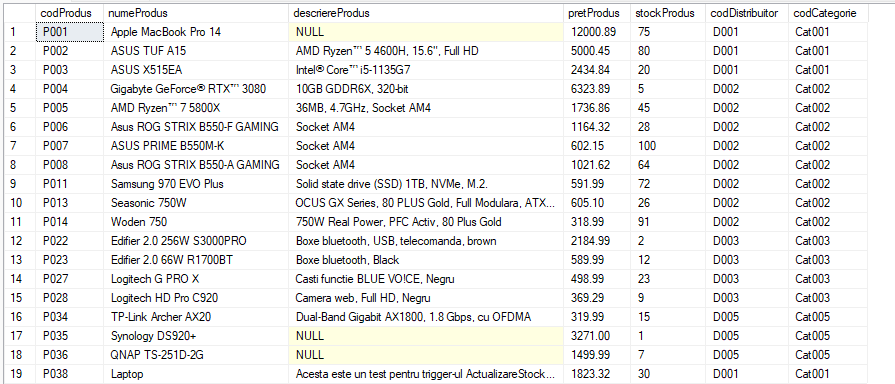
where codJudetClient = 'CJ';



--Selectati toate coloanele din tabelul tProdus pentru produsele care au pretul mai mare de 300

select \* from magazin.tProdus

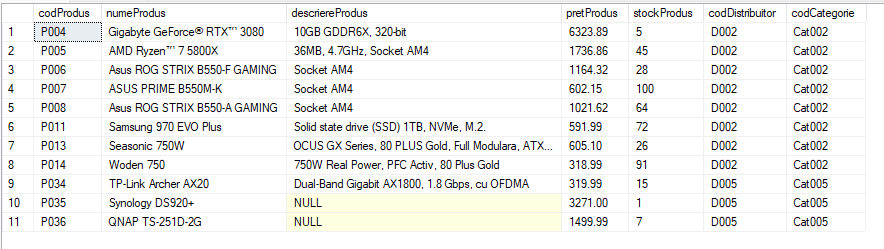
where pretProdus > 300;



--Selectati toate coloanele din tabelul tProdus unde pretul este mai mare de 250 si produsele fac parte fie din categoria 2, fie din categoria 5

select \* from magazin.tProdus

where pretProdus > 250 and (codCategorie = 'Cat002' or codCategorie = 'Cat005')



--Selectati distribuitori care incep cu litera "Q"

select \* from magazin.tDistribuitor

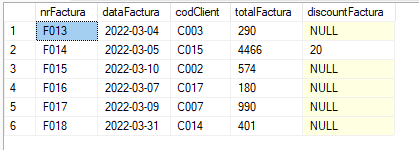
where numeDistribuitor like 'Q%'



--Selectati toate facturile din luna martie

select \* from magazin.tFactura

where dataFactura between '2022-03-01' and '2022-03-31'

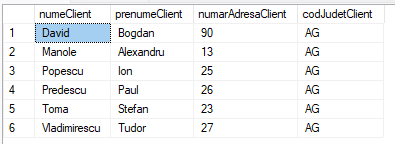


--Selecatri numarul adresei, judetul, numele si prenumele clientilor din judetul Arges si de asemenea, ordonatii in functie de nume

select numeClient, prenumeClient, numarAdresaClient, codJudetClient from magazin.tClient

where codJudetClient = 'AG' and numarAdresaClient is not null

order by numeClient

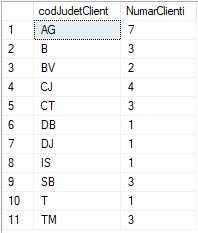


--Selectati cati clienti sunt din fiecare judet introdus in baza de date

select codJudetClient, count(codJudetClient) as NumarClienti

from magazin.tClient

group by codJudetClient



--Afisati cati clienti au acelasi numar al adresei, excluzand pe cei care au valoarea null

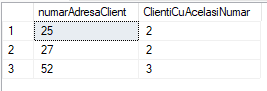
select numarAdresaClient, count(codClient) as ClientiCuAcelasiNumar

from magazin.tClient

where numarAdresaClient is not null

group by numarAdresaClient

having count(codClient) > 1

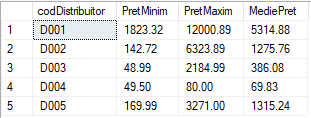


--Afisati pretul minim, maxim, si media tuturor produselor distribuitite de fiecare distribuitor in parte

select codDistribuitor, min(pretProdus) as PretMinim, max(pretProdus) as PretMaxim, cast(avg(pretProdus) as decimal(18,2)) as MediePret

from magazin.tProdus

group by codDistribuitor



Interogarea tabelelor din mai multe tabele

--Afisati toti clientii care au facut cel putin o comanda pe site

select A.codClient, A.numeClient, A.prenumeClient, B.nrFactura, B.totalFactura

from magazin.tClient as A

inner join

magazin.tFactura as B on A.codClient = B.codClient

order by A.codClient



--selectatii clientii care nu au facut nicio comanda

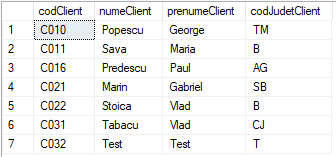
select A. codClient, A.numeClient, A.prenumeClient, A.codJudetClient

from magazin.tClient as A

left join

magazin.tFactura as B on A.codClient = B.codClient

where B.codClient is null



--Afisati clientii care au facut o comanda mai mare de 500 de lei

select A.codClient, A.numeClient, A.prenumeClient, B.totalFactura

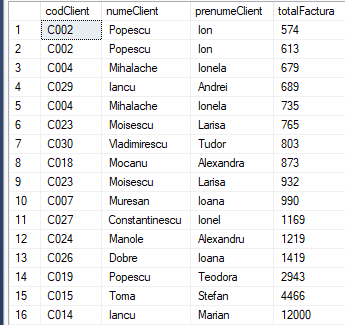
from magazin.tClient as A

inner join

magazin.tFactura as B on A.codClient = B.codClient

where B.totalFactura > 500

order by B.totalFactura



--afisati media facturilor in care codJudetClient = 'BV'

select A.codJudetClient, avg(B.totalFactura) as MedieFacturiBrasov

from magazin.tClient as A

inner join

magazin.tFactura as B on A.codClient = B.codClient

where A.codJudetClient = 'BV'

group by codJudetClient



--Afisati toate detaliile facturii clientului C014

select A.codClient, A.numeClient, A.prenumeClient, B.nrFactura, B.dataFactura, B.totalFactura, C.idVanzare, C.codProdus, C.cantitateVanduta

from magazin.tClient as A

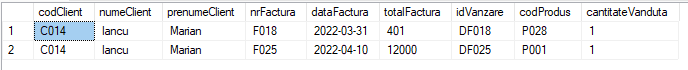
inner join

magazin.tFactura as B on A.codClient = B.codClient

inner join

magazin.tDetaliiFactura as C on B.nrFactura = C.nrFactura

where A.codClient = 'C014'



Subinterogari

--sa se afiseze numele prenumele clientului care a facut cea mai scumpa comanda

select numeClient, prenumeClient, totalFactura from magazin.tClient as A inner join magazin.tFactura as B

on A.codClient=B.codClient

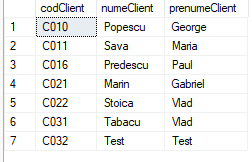
where totalFactura = (select max(totalFactura) from magazin.tFactura)



--sa se afiseze clientii care nu au facut nicio comanda pe site

select codClient, numeClient, prenumeClient from magazin.tClient

where codClient not in (select codClient from magazin.tFactura)

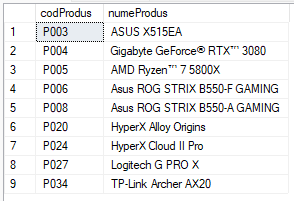


--Sa se afiseze produsele care contin "x" in denumirea lor

select codProdus, numeProdus

from magazin.tProdus

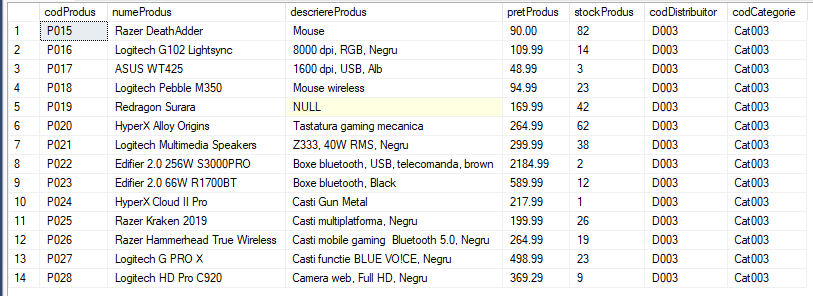
where numeProdus in (select numeProdus from magazin.tProdus where numeProdus like '%x%')



--selectati toate informatiile despre produsele care sunt distribuite de D003

select \* from magazin.tProdus p

where exists (select \* from magazin.tDistribuitor d where p.codDistribuitor = d.codDistribuitor and codDistribuitor = 'D003')



UNION INTERSECT EXCEPT

--Selectati produsele care au pretul mai mare de 300 lei si care, de asemenea, se afta in stock

select numeProdus, pretProdus, stockProdus

from magazin.tProdus

where pretProdus > 300

union

select numeProdus, pretProdus, stockProdus

from magazin.tProdus

where stockProdus is not null



--Selectacti clientii din Bucuresti si Cluj, folosind operatorul "UNION"

select numeClient, prenumeClient, codJudetClient

from magazin.tClient

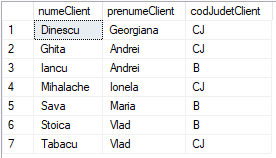
where codJudetClient = 'B'

union

select numeClient, prenumeClient, codJudetClient

from magazin.tClient

where codJudetClient = 'CJ'



--Afisati produsele care sunt distribuitite de D002 si nu sunt distribuite de D003 folosind operatorul "except"

select codProdus, numeProdus, codDistribuitor

from magazin.tProdus

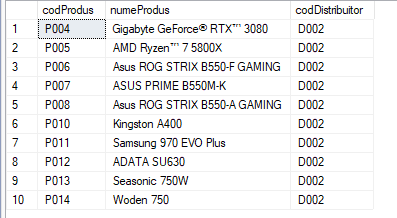
where codDistribuitor = 'D002'

except

select codProdus, numeProdus, codDistribuitor

from magazin.tProdus

where codDistribuitor = 'D003'



--Afisati clientii care au prenumele "Andrei" si sunt din Brasov, folosind operatorul "INTERSECT". In caz contrar, nu se va afisa nimic

select numeClient, prenumeClient, codJudetClient

from magazin.tClient

where prenumeClient like '%Andrei%'

intersect

select numeClient, prenumeClient, codJudetClient

from magazin.tClient

where codJudetClient = 'BV'



Case Statement

--Sa se determine facturile care au totalul mai mare de 450 de lei. In acest caz, factura va fi luata in considerare. In caz contrar, nu va fi.

select nrFactura, totalFactura,

case

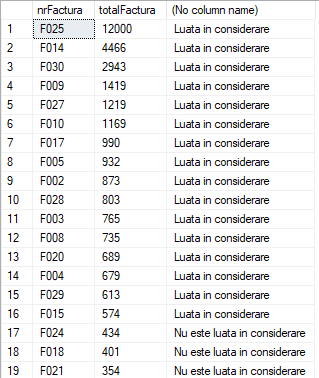
when totalFactura > 450 then 'Luata in considerare'

else 'Nu este luata in considerare'

END

FROM magazin.tFactura

order by totalFactura desc



--Selectati clientii care sunt din orasele mari ale tarii

select numeClient, prenumeClient, codJudetClient,

case

when codJudetClient = 'B' then 'Client din unul din orasele mari ale tarii'

when codJudetClient = 'CJ' then 'Client din unul din orasele mari ale tarii'

when codJudetClient = 'TM' then 'Client din unul din orasele mari ale tarii'

when codJudetClient = 'BV' then 'Client din unul din orasele mari ale tarii'

when codJudetClient = 'CT' then 'Client din unul din orasele mari ale tarii'

else 'Client care nu este din unul dintre orasele mari ale tarii'

END

from magazin.tClient

order by codJudetClient



Utilizarea vederilor

--Afisati clientii care au facut o comanda mai mare de 500 de lei

create view vClienti\_Peste\_500Lei

as

select A.codClient, A.numeClient, A.prenumeClient, B.totalFactura

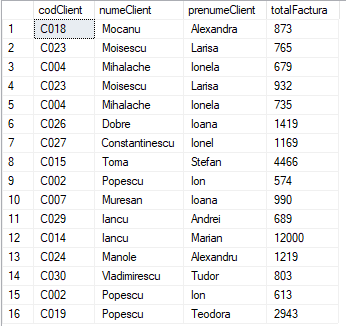
from magazin.tClient as A

inner join

magazin.tFactura as B on A.codClient = B.codClient

where B.totalFactura > 500

select \* from vClienti\_Peste\_500Lei



--Afisati toate detaliile facturii clientului C007

create view vDetalii\_Client\_C007

as

select A.codClient, A.numeClient, A.prenumeClient, B.nrFactura, B.dataFactura, B.totalFactura, C.idVanzare, C.codProdus, C.cantitateVanduta, D.numeProdus

from magazin.tClient as A

inner join

magazin.tFactura as B on A.codClient = B.codClient

inner join

magazin.tDetaliiFactura as C on B.nrFactura = C.nrFactura

inner join

magazin.tProdus as D on C.codProdus = D.codProdus

where A.codClient = 'C007'

select \* from vDetalii\_Client\_C007



--Afisati toate categoriile si toate produsele posibile

create view vCategorii\_si\_Produse\_Posibile

as

select A.codCategorie, A.numeCategorie, B.codProdus, B.numeProdus

from magazin.tCategorie as A

cross join

magazin.tProdus as B where A.codCategorie = B.codCategorie

select \* from vCategorii\_si\_Produse\_Posibile



Indexarea tabelelor

--creati index cu numele complet pentru fiecare client

create index ix\_NumeComplet on magazin.tClient(numeClient, prenumeClient)

--selectati toate datele din tabelul magazin.tClient folosind indexul ix\_NumeComplet

select \* from magazin.tClient with (index=ix\_NumeComplet)



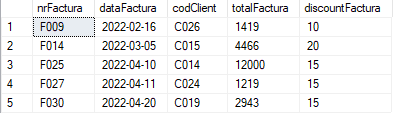
--creati un index denumit ix\_DetaliiFactura care sa cuprinda idVanzare, nrFactura, codProdus

create index ix\_DetaliiFactura on magazin.tDetaliiFactura(idVanzare, nrFactura, codProdus)

--afisati facturile care sunt in valoare de peste 1200 lei, folosind indexul ix\_TotaldataFactura

select \* from magazin.tFactura with (index=ix\_TotalDataFactura)

where totalFactura > 1200



--creati un index denumit TotalDataFactura care sa cuprinda nrFactura, dataFactura, totalFactura

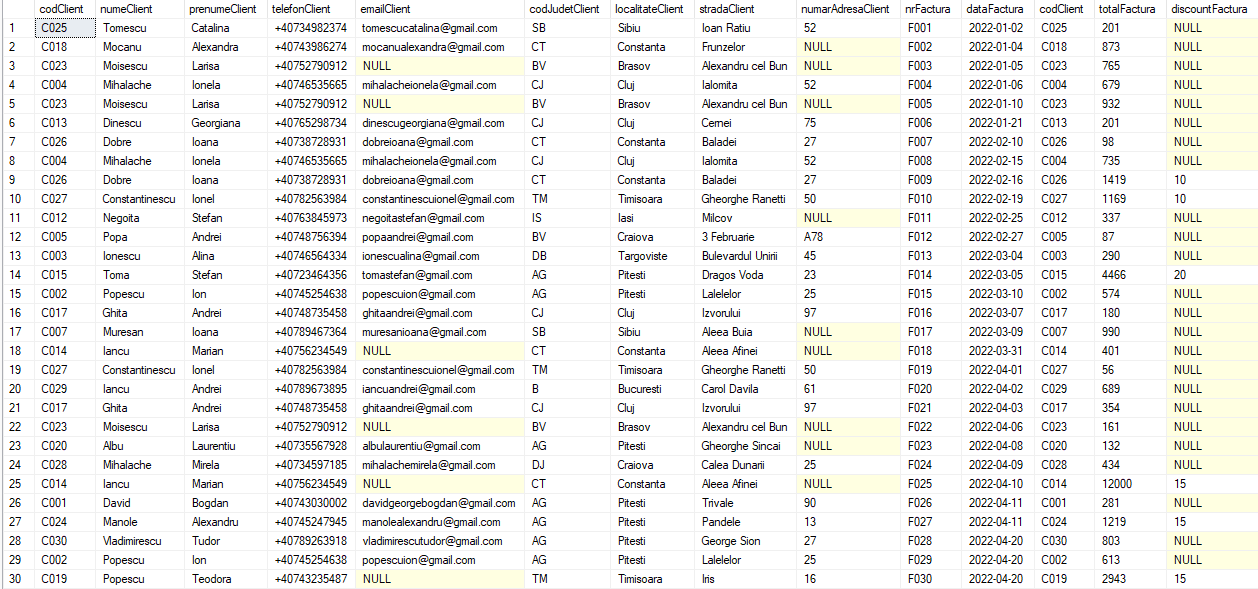
create index ix\_TotalDataFactura on magazin.tFactura(nrFactura, dataFactura, totalFactura)

--selectati toti clientii care apar pe cel putin o factura, folosind indexul ix\_NumeComplet

select \* from magazin.tClient as A with (index=ix\_NumeComplet)

inner join

magazin.tFactura as B on A.codClient=B.codClient



Sinteza datelor (Rollup & Cube)

Operatorul ROLLUP crează subtotaluri şi statistici care pot fi agregate de la cel mai detaliat nivel către un total general: la început se calculează agregatele standard specificate în clauza GROUP BY, apoi se creează subtotaluri la niveluri ascendente începând de la dreapta către stânga în lista expresiilor specificate în group by, în final se creează un total general.

select codProdus, numeProdus,

sum(pretProdus) as [Total preturi],

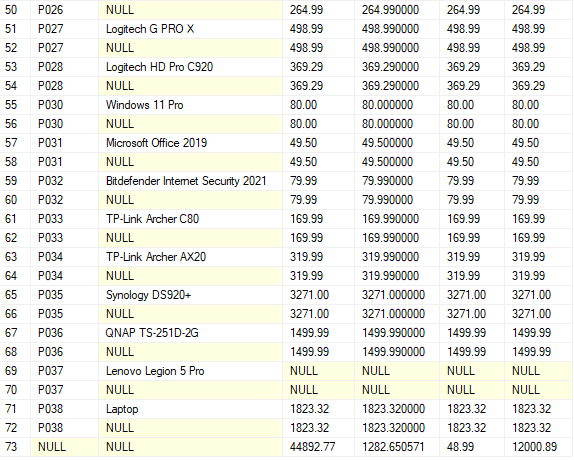
avg(pretProdus) as [Pret Mediu],

min(pretProdus) as [Pret Minim],

max(pretProdus) as [Pret Maxim]

from magazin.tProdus

group by codProdus, numeProdus with rollup



Operatorul CUBE produce subtotaluri pentru toate combinaţiile posibile de grupări specificate în clauza GROUP BY, precum şi un total general.

select codProdus, numeProdus,

sum(pretProdus) as [Total preturi],

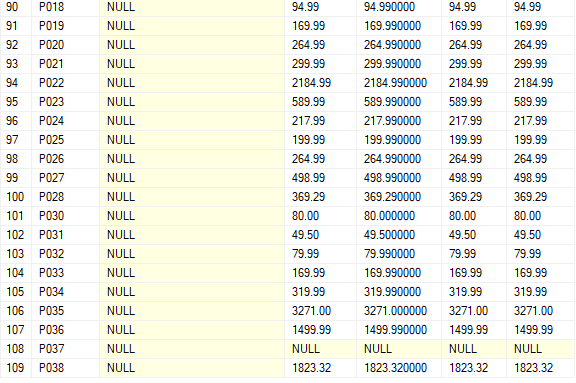
avg(pretProdus) as [Pret Mediu],

min(pretProdus) as [Pret Minim],

max(pretProdus) as [Pret Maxim]

from magazin.tProdus

group by codProdus, numeProdus with cube



Limbajul de manipulare al datelor

--Stergeti datele clientului care are codul C006

delete from magazin.tClient

where codClient = 'C033'

--Actualizati clientului C005 codul judetului in BV

update magazin.tClient

set codJudetClient = 'BV'

where codClient = 'C005'

Functii & Proceduri

--Sa se elaboreze o functie Transact SQL, pentru determinarea numarului de clienti dintr-un judet prin parametrul de intrare @codJudetClient

go

create function magazin.NrClientiJudet(@codJudetClient char(2))

returns int

begin

declare @nrClienti int

set @nrClienti = (select count(codClient) from magazin.tClient where codJudetClient = @codJudetClient)

return @nrClienti

end

select magazin.NrClientiJudet('AG') as NumarClientiJudet



--Sa se elaboreze o functie Transact SQL, pentru determinarea numelui clientului care are nrFactura specificat prin parametrul de intrare @nrFactura

go

alter function magazin.NumeClient(@nrFactura char(10))

returns varchar(60)

begin

declare @numeClient varchar(60)

set @numeClient = (select A.numeClient+' '+A.prenumeClient as NumeComplet from magazin.tClient as A inner join magazin.tFactura as B on A.codClient = B.codClient where B.nrFactura = @nrFactura)

return @numeClient

end

select magazin.NumeClient('F003') as NumeClient



--Creati o procedura care afiseaza produsele

create procedure sp\_Produse @codProdus char(10) = '%', @numeProdus varchar(60)='%'

as

begin

select codProdus, numeProdus from magazin.tProdus

where codProdus like @codProdus and numeProdus like @numeProdus

end



--Creati o procedura care actualizeaza pretul unui produs dat ca parametru

create procedure sp\_ActualizarePret

@codProdus char(10), @pretProdus decimal(18,2)

as

begin

begin try

if exists(select codProdus, pretProdus from magazin.tProdus where codProdus = @codProdus and pretProdus = @pretProdus)

update magazin.tProdus set pretProdus = @pretProdus where codProdus = @codProdus

else

insert into magazin.tProdus (codProdus, pretProdus) values (@codProdus, @pretProdus)

return 1;

end try

begin catch

print ('Date eronate');

return 0;

end catch

end

--Elaborati o procedura pentru calculul numarului de produse si pentru calculul pretului mediu al produselor dintr-o categorie

create procedure sp\_InfoProducts @nrProdus int out, @pretMediu numeric(7,2) out, @categorie char(10)

as

begin

select @nrProdus = count(codProdus), @pretMediu = avg(pretProdus) from magazin.tProdus where codCategorie = @categorie

end

declare @nrP int, @Pm numeric(10,2)

execute sp\_InfoProducts @nrP out, @Pm out, 'Cat003'

select @nrP as [Numar Produse], @nrP as [Pret Mediu]



Tranzactii

create procedure sp\_VanzareProdus

@codProdus char(10),

@cantitateVanduta int

as

begin

--Verificam stock-ul valabil pentru produsele pe care vrem sa le vindem

declare @stockValabil int

select @stockValabil = stockProdus from magazin.tProdus where codProdus = @codProdus

--Daca stock-ul nu este suficient

if(@stockValabil < @cantitateVanduta)

begin

Raiserror('Stock insuficient',16,1)

end

else --Daca stock-ul este suficient

begin

begin tran

--Reducem stock-ul

update magazin.tProdus set stockProdus = (stockProdus - @cantitateVanduta)

where codProdus = @codProdus

declare @maxIdVanzare char(10)

select @maxIdVanzare = case when max(idVanzare) is null

then 0

else max(idVanzare) end

from magazin.tDetaliiFactura

set @maxIdVanzare = @maxIdVanzare + 1

insert into magazin.tDetaliiFactura values (@maxIdVanzare, @codProdus, @cantitateVanduta)

commit tran

end

end

--Sa se implementeze un cursor care afiseaza toti clientii din Arges

declare @codJudet char(2) = 'AG', @i int = 1

declare cursorClientiArges cursor scroll

for select codClient, numeClient, PrenumeClient, emailClient, localitateClient from magazin.tClient where codJudetClient = @codJudet

open cursorClientiArges

declare @codClient char(10), @numeClient varchar(60), @prenumeClient varchar(60), @emailClient varchar(60), @localitateClient varchar(60)

while @@FETCH\_STATUS = 0

begin

print @codClient+' '+@numeClient+' '+@prenumeClient+' '+@emailClient+' '+@localitateClient

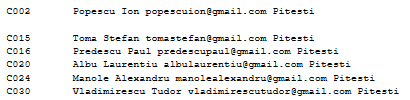
set @i+=1

fetch absolute @i from cursorClientiArges into @codClient, @numeClient, @prenumeClient, @emailClient, @localitateClient

end

close cursorClientiArges

deallocate cursorClientiArges



--Sa se implementeze un cursor care sa afiseze toti clientii cu toate facturile lor

declare @codClient char(10), @nrFactura char(10)

declare cursorClientiFacturi cursor for

select A.codClient, B.nrFactura from magazin.tClient as A inner join magazin.tFactura as B on A.codClient = B.codClient

open cursorClientiFacturi

fetch next from cursorClientiFacturi into @codClient, @nrFactura

while @@FETCH\_STATUS = 0

begin

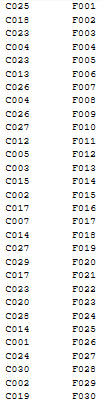
print @codClient+' '+@nrFactura+' '

fetch next from cursorClientiFacturi into @codClient, @nrFactura

end

close cursorClientiFacturi

deallocate cursorClientiFacturi



Triggers

create trigger magazin.tg\_ActualizareStockProduse

on magazin.tProdus

for insert, update, delete

as

begin

set nocount on

begin try

declare @codProdus char(10)

if exists (select \* from inserted)

declare crs cursor for select codProdus from inserted

else

declare crs cursor for select codProdus from deleted

open crs

fetch next from crs into @codProdus

declare @numeProdus varchar(60), @descriereProdus varchar(500), @pretProdus decimal(18,2), @stockProdus int, @codDistribuitor char(10), @codCategorie char(10)

while @@FETCH\_STATUS = 0

begin

set @numeProdus = (select numeProdus from magazin.tProdus where codProdus = @codProdus)

set @descriereProdus = (select descriereProdus from magazin.tProdus where codProdus = @codProdus)

set @pretProdus = (select pretProdus from magazin.tProdus where codProdus = @codProdus)

set @stockProdus = (select stockProdus from magazin.tProdus where codProdus = @codProdus)

set @codDistribuitor = (select codDistribuitor from magazin.tProdus where codProdus = @codProdus)

set @codCategorie = (select codCategorie from magazin.tProdus where codProdus = @codProdus)

update magazin.tProdus

set numeProdus = @numeProdus where codProdus=@codProdus

update magazin.tProdus

set descriereProdus = @descriereProdus where codProdus =@codProdus

update magazin.tProdus

set pretProdus = @pretProdus where codProdus= @codProdus

update magazin.tProdus

set stockProdus = @stockProdus where codProdus = @codProdus

update magazin.tProdus

set codDistribuitor = @codDistribuitor where codProdus = @codProdus

update magazin.tProdus

set codCategorie = @codCategorie where codProdus = @codProdus

fetch next from crs into @codProdus

end

close crs

deallocate crs

end try

begin catch

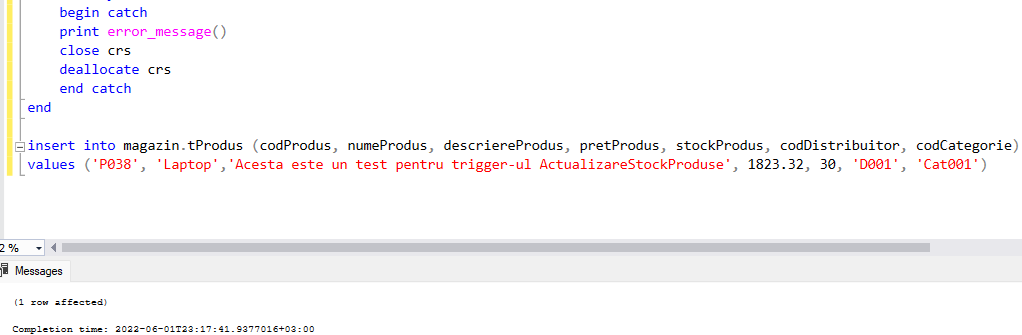
print error\_message()

close crs

deallocate crs

end catch

end





--Sa se implementeze un trigger care tine evidenta clientilor inregistrati pe site

create table magazin.tIstoric\_Clienti

(

id int not null identity (1,1) primary key,

codClient char(10) not null,

numeClient varchar(60) not null,

prenumeClient varchar(60) not null,

telefonClient char(15) not null,

emailClient varchar(60),

codJudetClient char(2) not null,

localitateClient varchar(60) not null,

stradaClient varchar(60) not null,

numarAdresaClient char(5),

statusClient char(10) not null,

check (statusClient = 'Inserat' or statusClient = 'Sters')

)

create trigger magazin.tg\_EvidentaClienti

on magazin.tClient

after insert, delete

as

begin

set nocount on;

insert into magazin.tIstoric\_Clienti(codClient, numeClient, prenumeClient, telefonClient, emailClient, codJudetClient, localitateClient, stradaClient, numarAdresaClient, statusClient)

select A.codClient, A.numeClient, A.prenumeClient, A.telefonClient, A.emailClient, A.codJudetClient, A.localitateClient, A.stradaClient, A.numarAdresaClient, 'Inserat' from inserted as A

union

select B.codClient, B.numeClient, B.prenumeClient, B.telefonClient, B.emailClient, B.codJudetClient, B.localitateClient, B.stradaClient, B.numarAdresaClient, 'Sters' from deleted as B

end

insert into magazin.tClient

values ('C032', 'Test', 'Test','Test', 'Test', 'T', 'Test', 'Test', 32)

delete from magazin.tClient

where codClient = 'C008'

