**Magazin Online de**

**Calculatoare si**

**Componente**

**Student: Cuciurean Catalin**

**Grupa : 30244**

Contents

[**1.** **Descrierea proiectului** 3](#_Toc42201717)

[**2.** **Diagrama Bazei de Date** 4](#_Toc42201718)

[**3.** **Constrangeri** 5](#_Toc42201719)

[a. Constrangeri de unicitate 5](#_Toc42201720)

[b. Constrangeri de integritate 6](#_Toc42201721)

[**4.** **Crearea Tabele** 6](#_Toc42201722)

[**5.** **Populare Tabele** 7](#_Toc42201723)

[**6.** **Vederi** 8](#_Toc42201724)

[1. Afisarea contului la banca a unui client 9](#_Toc42201725)

[2. Afisare produse cu pretul mai mare de 2000 de lei 9](#_Toc42201726)

[**7.** **Proceduri Stocate** 9](#_Toc42201727)

[1. Afisarea numelui complet al administartorului, avand numele dat ca parametru 10](#_Toc42201728)

[2. Adaugarea unui nou cont la banca 10](#_Toc42201729)

[**8.** **Trigger** 11](#_Toc42201730)

[1. Inserare Administrator 11](#_Toc42201731)

[2. Stergere Administrator 11](#_Toc42201732)

[3. Validare Pret 11](#_Toc42201733)

[9. Rapoarte 12](#_Toc42201734)

[a) Raport simplu 12](#_Toc42201735)

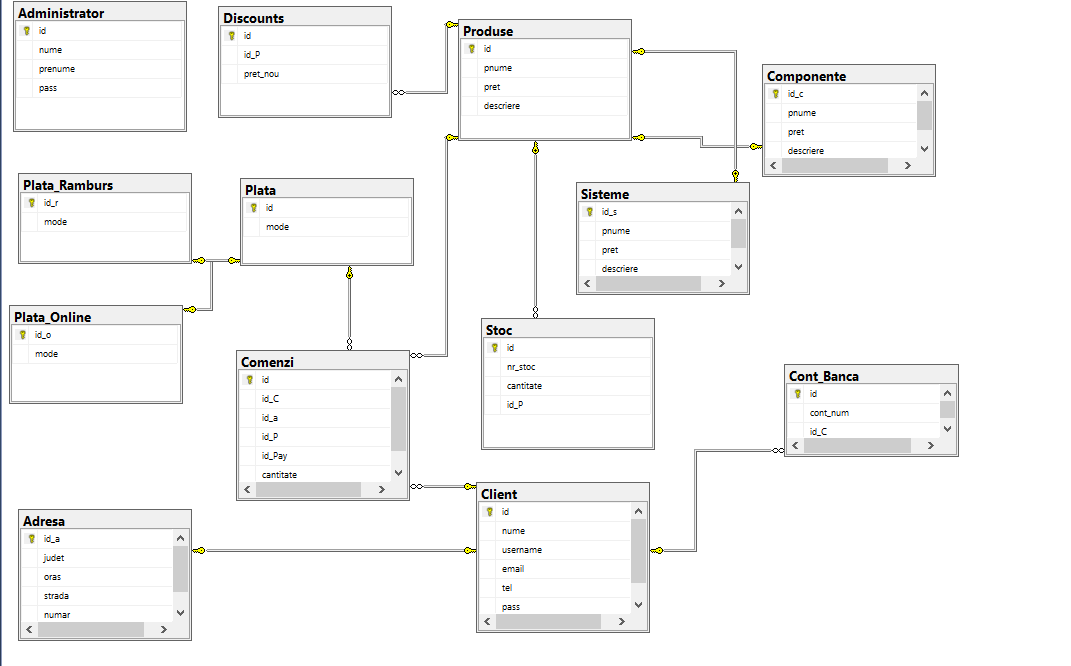
[b) Raport cu parametru 12](#_Toc42201736)

[c) Raport cu imagini 13](#_Toc42201737)

# **Descrierea proiectului**

Proiectul reprezinta crearea unui magazin online de calculatoare/PC si diferite componenete pentru acestea. Acest proiect va contine administratori ai magazinului, clienti care vor putea face diferite comenzi, iar magazinul va contine diferite modele de calculatoare si diferite modele de componente. Acest proiect are ca scop consolidarea cunostintelor dobandite in cadrul laboratorului de proiectare a bazelor de date.

# **Diagrama Bazei de Date**



# **Constrangeri**

## Constrangeri de unicitate

Constrangerile de unicitate reprezinta un index unic. Acestea au fost aplicate la crearea tabellor pentu parametri care necesita unicitate.

Exemplu:

CREATE TABLE Administrator

(id INT NOT NULL ,

nume varchar(50) unique,

prenume varchar(30) unique,

pass varchar(25),

PRIMARY KEY(id)

);

Pentru tabela administrator, numele si prenumele sunt unice.

## Constrangeri de integritate

Atunci cand se face o stergere sau o modificare intr-o tabela, aceasta sa fie vizibila si in tabelele cu care are relatii. Astfel am ales sa aplic operatia cascade pentru diferite tabele.

Exemplu:

CREATE TABLE Sisteme

(id\_s INT NOT NULL ,

pnume VARCHAR(50) NOT NULL unique,

pret float,

descriere varchar(5000),

PRIMARY KEY(id\_s),

CONSTRAINT Sisteme\_produse

FOREIGN KEY (id\_s)

REFERENCES Produse(id)

ON DELETE CASCADE

ON UPDATE CASCADE

);

La crearea tabelei sisteme, atunci cand se face o stergere sau o modificare in tabela, aceasta va fi vizibila si in tabela produse.

# **Crearea Tabele**

Am adaugat doua exemple pentru crearea tabelelor. Tabele sunt create la executia codului cu atributele specifice fiecareia.

CREATE TABLE Administrator

(id INT NOT NULL ,

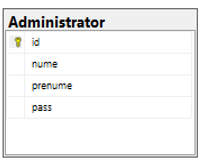
nume varchar(50) unique,

prenume varchar(30) unique,

pass varchar(25),

PRIMARY KEY(id)

);



CREATE TABLE Sisteme

(id\_s INT NOT NULL ,

pnume VARCHAR(50) NOT NULL unique,

pret float,

descriere varchar(5000),

PRIMARY KEY(id\_s),

CONSTRAINT Sisteme\_produse

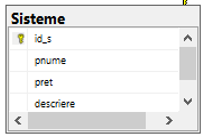
FOREIGN KEY (id\_s)

REFERENCES Produse(id)

ON DELETE CASCADE

ON UPDATE CASCADE

);



# **Populare Tabele**

Urmatorul cod reprezinte inserarea diferitelor inregistrari in tabele. Se alege tabele si atributele care trebuie asaugate, apoi de definesc diferite valori pentru atributele respective.

INSERT INTO Administrator (id,nume,prenume,pass) VALUES

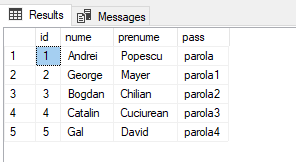
(1,'Andrei','Popescu','parola'),

(2,'George','Mayer','parola1'),

(3,'Bogdan','Chilian','parola2'),

(4,'Catalin','Cuciurean','parola3'),

(5,'Gal','David','parola4');



# **Vederi**

## Afisarea contului la banca a unui client

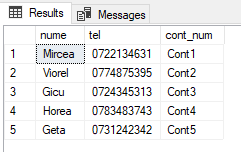
CREATE VIEW A\_Cont AS

SELECT dbo.Client.nume, dbo.Client.tel, dbo.Cont\_Banca.cont\_num

FROM dbo.Client INNER JOIN

dbo.Cont\_Banca ON dbo.Client.id = dbo.Cont\_Banca.id\_C

SELECT \* FROM A\_Cont



## Afisare produse cu pretul mai mare de 2000 de lei

CREATE VIEW A\_Produse

AS

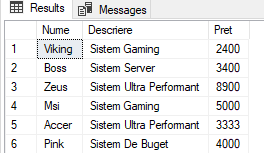
SELECT Produse.pnume As Nume, Produse.descriere As Descriere, Produse.pret as Pret

From Produse

Where pret > 2000

Go

SELECT \* FROM A\_Produse



# **Proceduri Stocate**

## Afisarea numelui complet al administartorului, avand numele dat ca parametru

CREATE PROCEDURE Selectare\_Administratori (@NumeAdmin nvarchar(50))

AS

Begin

SELECT distinct a.nume +' '+ a.prenume as 'Nume Administrator'

FROM

Administrator a

Where a.nume like @NumeAdmin

END

GO

EXECUTE Selectare\_Administratori'%Andrei%';



## Adaugarea unui nou cont la banca

CREATE PROCEDURE AdaugareContBanca @id\_C int, @cont\_nume varchar(40)

AS

BEGIN

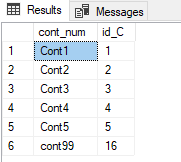
Set NOCOUNT ON;

Insert INTO Cont\_Banca(id\_C, cont\_num)

Values(@id\_C, @cont\_nume)

End

EXECUTE AdaugareContBanca 16,'cont99';



# **Trigger**

## Inserare Administrator

CREATE OR ALTER TRIGGER Adaugare\_nume\_admin

ON Administrator

After Insert

AS

DECLARE @nume varchar(4000);

DECLARE @prenume varchar(4000);

Select @nume=inserted.nume From inserted;

Select @prenume=inserted.prenume From inserted;

BEGIN

RAISERROR('Nume incorect',50,1);

ROLLBACK;

END

## Stergere Administrator

CREATE OR ALTER TRIGGER Stergere\_admin

ON Administrator

FOR DELETE

AS BEGIN SET NOCOUNT ON;

DECLARE @nume varchar(4000);

DECLARE @prenume varchar(4000);

SELECT @nume=DELETED.nume FROM DELETED;

SELECT @prenume=DELETED.prenume FROM DELETED;

BEGIN

RAISERROR('Nu se poate sterge',50,1);

ROLLBACK;

END

END;

### Validare Pret

CREATE TRIGGER Validare\_pret ON produse

FOR Insert

As

BEGIN

IF ( Select pret from Produse ) <= 0

Declare @Message2 varchar(4000);

SET @Message2 = 'Pretul nu poate fi mai mic ca 0';

END

# Rapoarte

## Raport simplu

SELECT Comenzi.id, Comenzi.id\_C, Comenzi.id\_a, Comenzi.id\_P, Comenzi.id\_Pay, Comenzi.cantitate,

Client.username, Client.email, Client.tel, Adresa.judet, Adresa.oras, Adresa.strada, Adresa.numar,

Produse.pnume, Produse.pret,

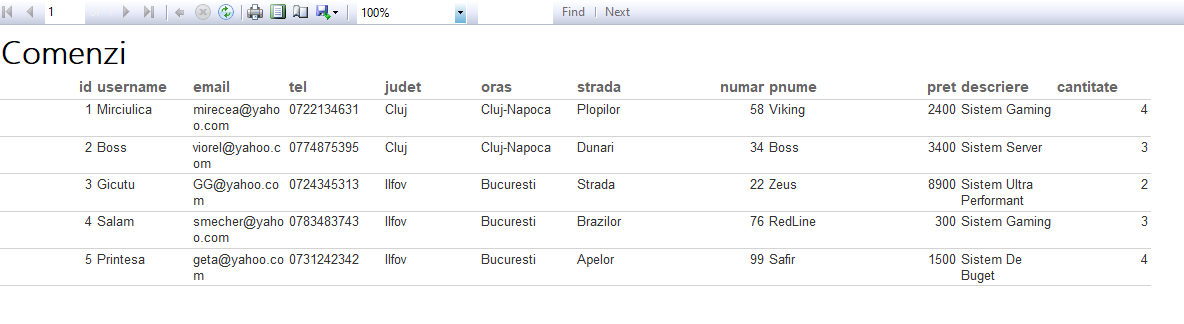
Produse.descriere

FROM Comenzi INNER JOIN

Client ON Comenzi.id\_C = Client.id INNER JOIN

Adresa ON Client.id = Adresa.id\_a INNER JOIN

Produse ON Comenzi.id\_P = Produse.id



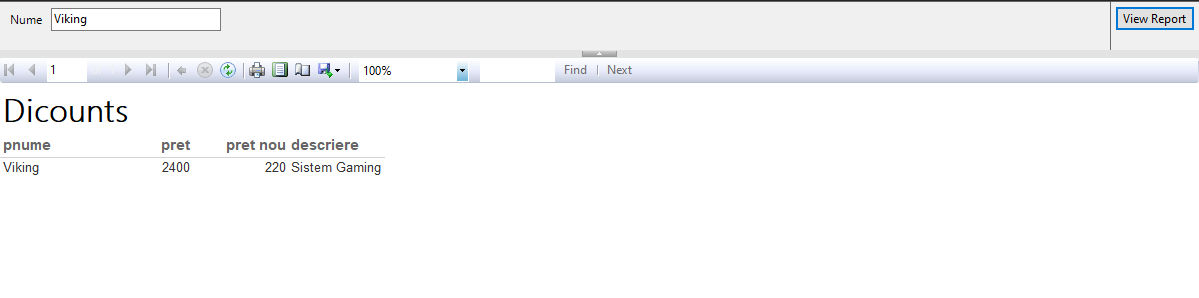
## Raport cu parametru

SELECT Produse.pnume, Produse.pret, Discounts.pret\_nou, Produse.descriere

FROM Produse INNER JOIN

Discounts ON Produse.id = Discounts.id\_P

Where Produse.pnume = @Nume



## Raport cu imagini

SELECT Produse.pnume, Produse.pret, Produse.descriere, Stoc.cantitate, Stoc.nr\_stoc

FROM Stoc INNER JOIN

Produse ON Stoc.id\_P = Produse.id

