UNIVERSITATEA POLITEHNICA BUCUREȘTI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

|  |  |
| --- | --- |
| upb | cs |

MAGAZIN ONLINE DE CEASURI

Emilia-Mihaela Bögözi

BUCUREŞTI

2021

**CUPRINS**

[1 Introducere 2](#_Toc59887752)

[2 Descrierea temei 2](#_Toc59887753)

[3 Descrierea bazei de date 3](#_Toc59887754)

[3.1 Diagrama bazei de date 3](#_Toc59887755)

[3.2 Structura tabelelor 4](#_Toc59887756)

[3.3 Descrierea constrângerilor de integritate 5](#_Toc59887757)

[3.4 Descrierea procedurilor și funcțiilor 6](#_Toc59887758)

[4 Descrierea aplicației 7](#_Toc59887759)

[4.1 Diagrama si structura claselor 7](#_Toc59887760)

[4.2 Diagrama de stări și fluxul de lucru (workflow) pentru aplicație 7](#_Toc59887761)

[4.3 Prezentarea modului în care se face conexiunea cu baza de date 7](#_Toc59887762)

[5 Capturi de ecran pentru interfețe și rapoarte 8](#_Toc59887763)

[6 Concluzii 9](#_Toc59887764)

[7 Bibliografie 9](#_Toc59887765)

# Introducere

De-a lungul timpului comerțul electronic a caștigat rapid teren ca paradigmă de afaceri acceptată și utilizată. Din ce în ce mai multe afaceri implementează site-uri web care ofera funcționalități pentru realizarea cumpărăturilor web. [1] Obiectivul acestui proiect este de a dezvolta un magazin online cu ceasuri, în care produsele pot fi cumpărate în confortul casei prin Internet.

Un magazin online este un magazin virtual de pe Internet unde clienții pot naviga în catalog și pot selecta produse de interes propriu. Articolele selectate pot fi colectate într-un coș de cumpărături. Acestea pot fi adăugate sau eliminate, în final comanda putând fi trimisă.

# Descrierea temei

* Initial, la pornirea aplicației, se deschide pagina Acasă, iar în colțul aplicației este prezentat un label „Neconectat”, care notifică utilizatorul că acesta nu este logat.
* Utilizatorul poate alege din urmatoarele variante:
* Să vadă produsele puse la dispoziție de magazin
* Să completeze un formular de contact pentru trimiterea mesajelor către admini cu scopul rezolvării unor probleme sau pentru răspunderea întrebărilor clienților; în acest caz trebuie să completeze niște câmpuri precum:
* Nume
* Prenume
* Email
* Telefon
* Mesaj
* Să își creeze un cont nou
* Să se logheze cu un cont deja existent
* Dacă utilizatorul alege să își creeze un cont nou, atunci acesta va trebui să completeze niște câmpuri, precum:
* Username
* Parolă
* Email
* Telefon
* După crearea unui cont nou:
* dacă datele sunt valide, va apărea mesajul „Contul a fost creat cu succes!”
* dacă userul deja există, va apărea mesajul „Cont existent”
* După crearea unui cont nou, utilizatorul nu este automat conectat, de aceea trebuie să se conecteze
* Dacă utilizatorul alege să se conecteze, atunci va trebui să completeze urmatoarele câmpuri:
* Username
* Parola
* După conectare:
* Dacă datele au fost introduse corect, va apărea mesajul: „Conectat!”
* Dacă datele introduse nu sunt valide, va apărea mesajul „Username sau parolă invalid/e”
* Dacă datele introduse la conectare sunt valide și utilizatorul este conectat, atunci în colțul aplicației labelul „Neconectat” devine numele utilizatorului ce s-a conectat
* După conectare, utilizatorul va avea mai multe opțiuni, precum:
* În pagina cu produse va apărea un buton de “Adaugă în coș”, astfel utilizatorul poate selecta un produs și îl poate adăuga în coș
* Apare un buton „Coșul meu” în care poți verifica produsele adăugate în coș
* Langa butonul „Cosul meu” va aparea un label cu numarul produselor din cos
* În „Coșul meu” apar produsele adăugate în coș, având posibilitatea de a elimina produse prin selectarea produsului ce se dorește a fi eliminat și apăsarea butonului „Elimină produsul din coș” sau de a finaliza comanda prin apăsarea butonului „Finalizează comanda”
* Pentru deconectare se apasă butonul „Deconectare” care redenumește label-ul din colțul aplicației în „Neconectat” și elimină opțiunea de „Coșul meu”
* Utilizatorii care vizitează aplicația Desktop pot realiza următoarele acțiuni:
* Își pot crea un cont nou
* Se pot conecta la aplicație
* Pot vizualiza produsele puse la dispoziție de magazin
* Pot completa un formular de contact pentru trimiterea mesajelor către admini cu scopul rezolvării unor probleme sau pentru răspunderea întrebărilor clienților
* Utilizatorii conectați în aplicația Desktop pot realiza următoarele acțiuni:
* Pot vizualiza produsele puse la dispoziție de magazin
* Pot adăuga în coșul de cumpărături produsele dorite
* Pot vizualiza coșul de cumpărături, împreună cu suma produselor aflate în acesta
* Pot realiza comanda produselor aflate în coșul de cumpărături
* Se pot deconecta din cont

# Descrierea bazei de date

## Diagrama bazei de date

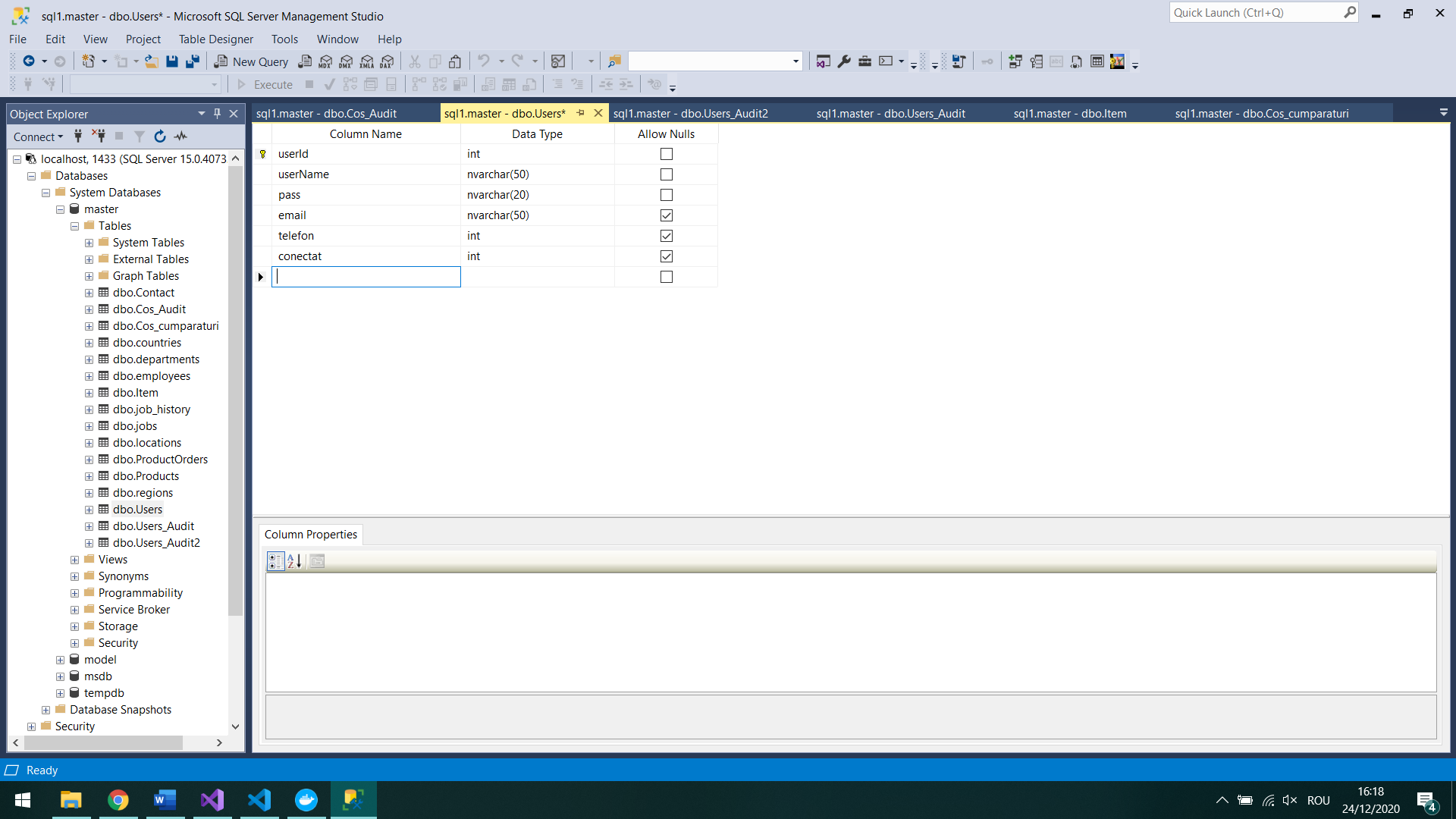
Graphical user interface, application

Description automatically generated

Figură ‑: Diagrama baze de date

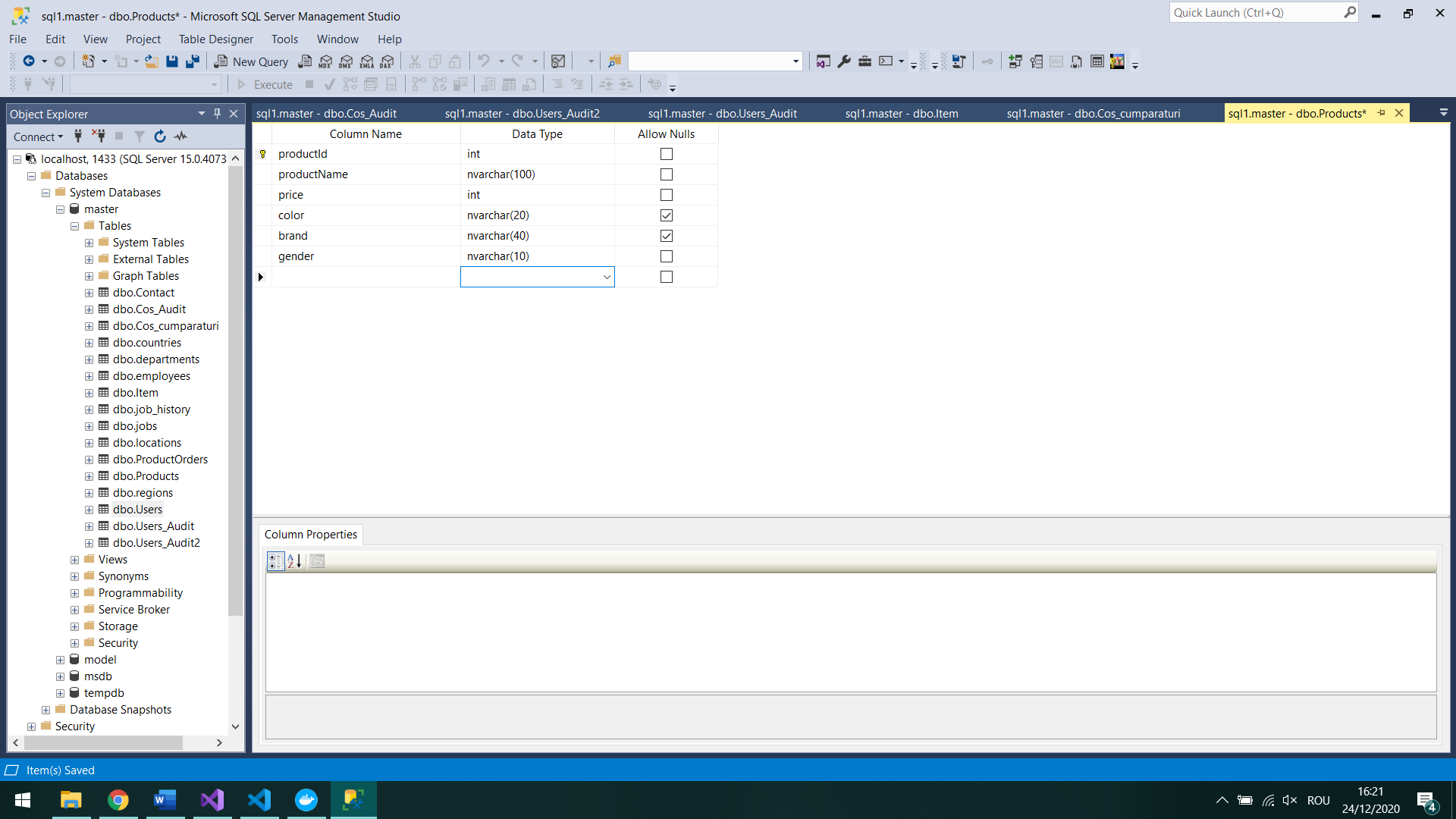
## Structura tabelelor

1. Users



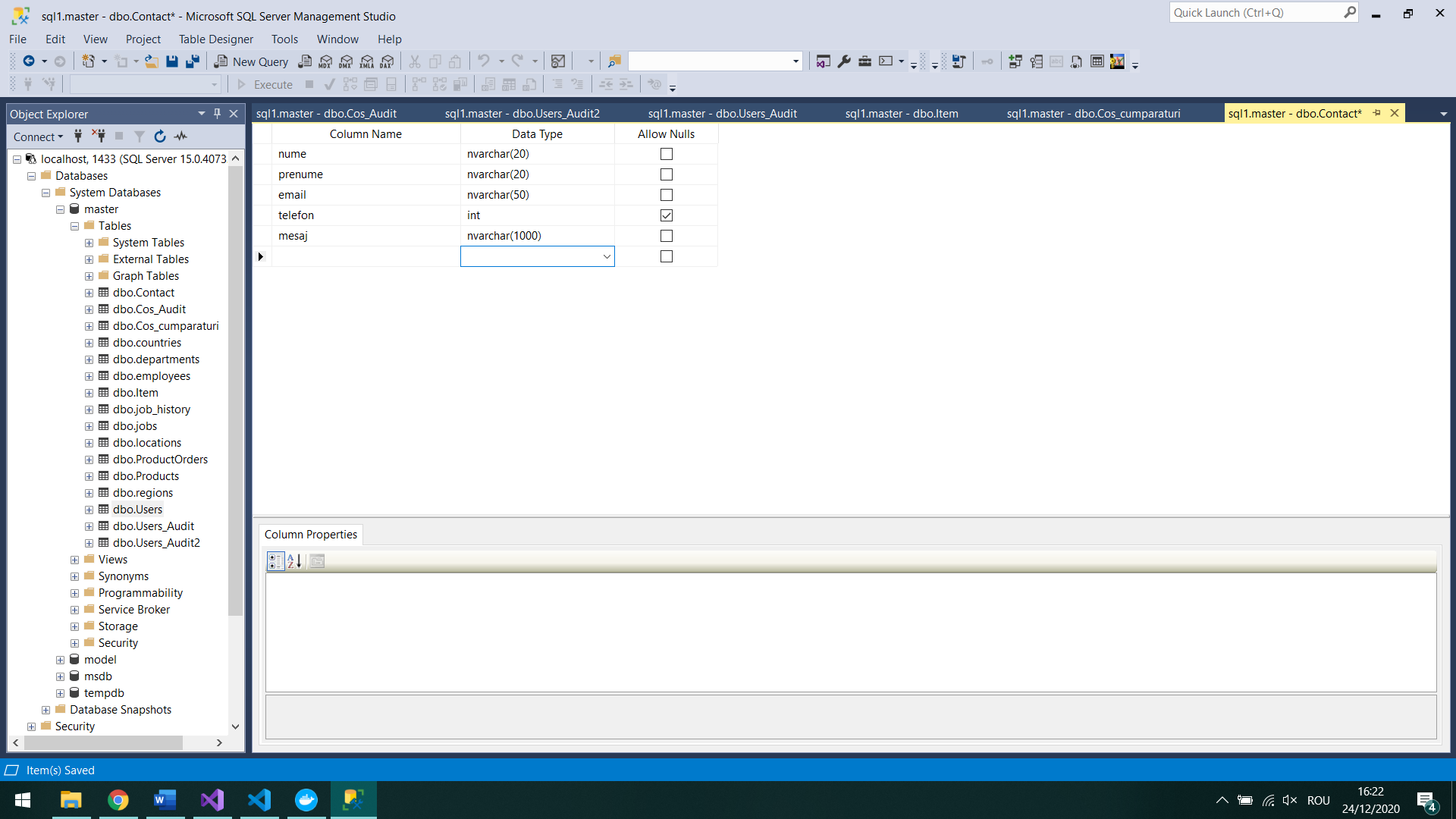
Figură ‑: Tabel Users

1. Products



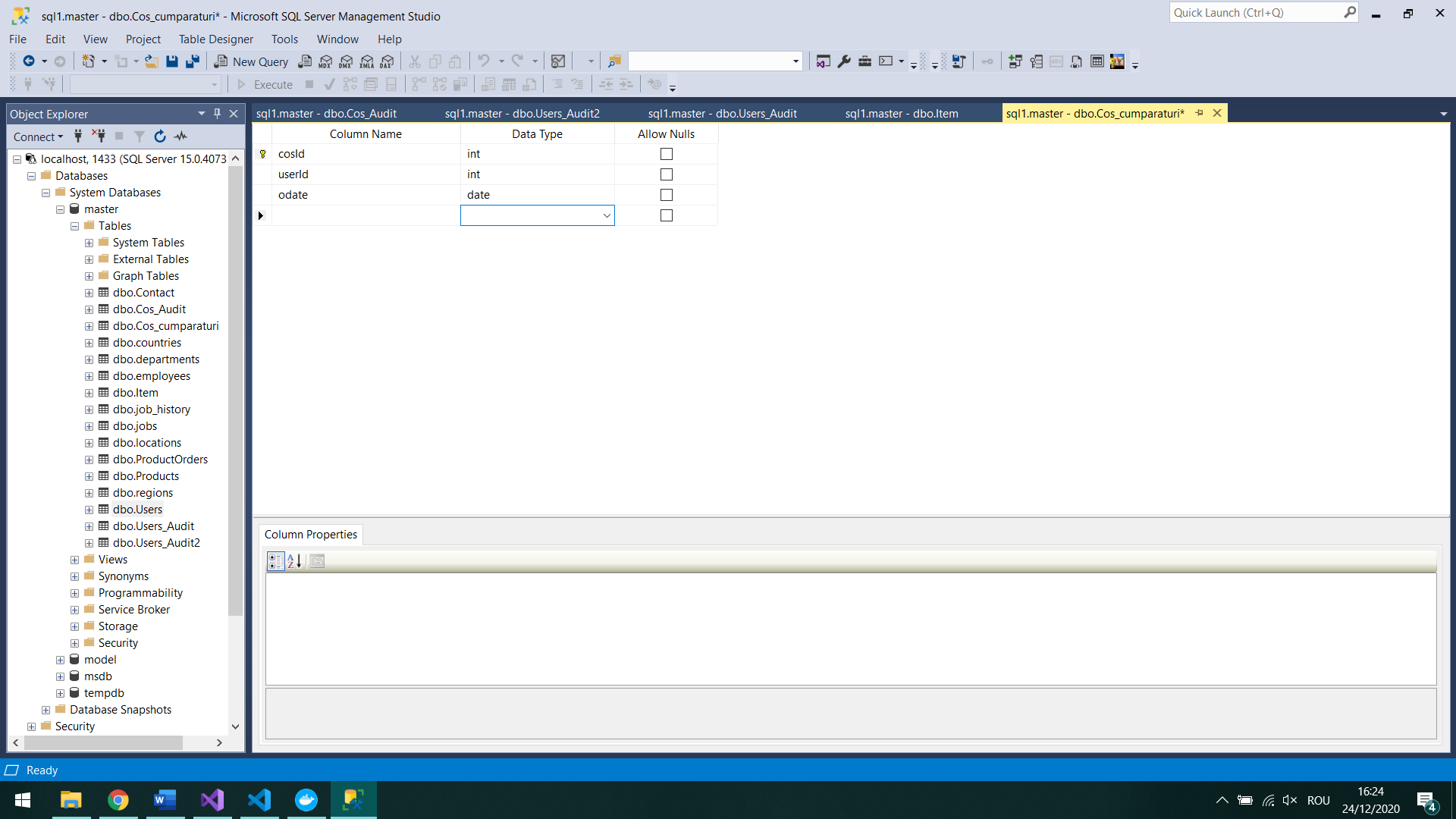
Figură ‑: Tabel Products

1. Contact



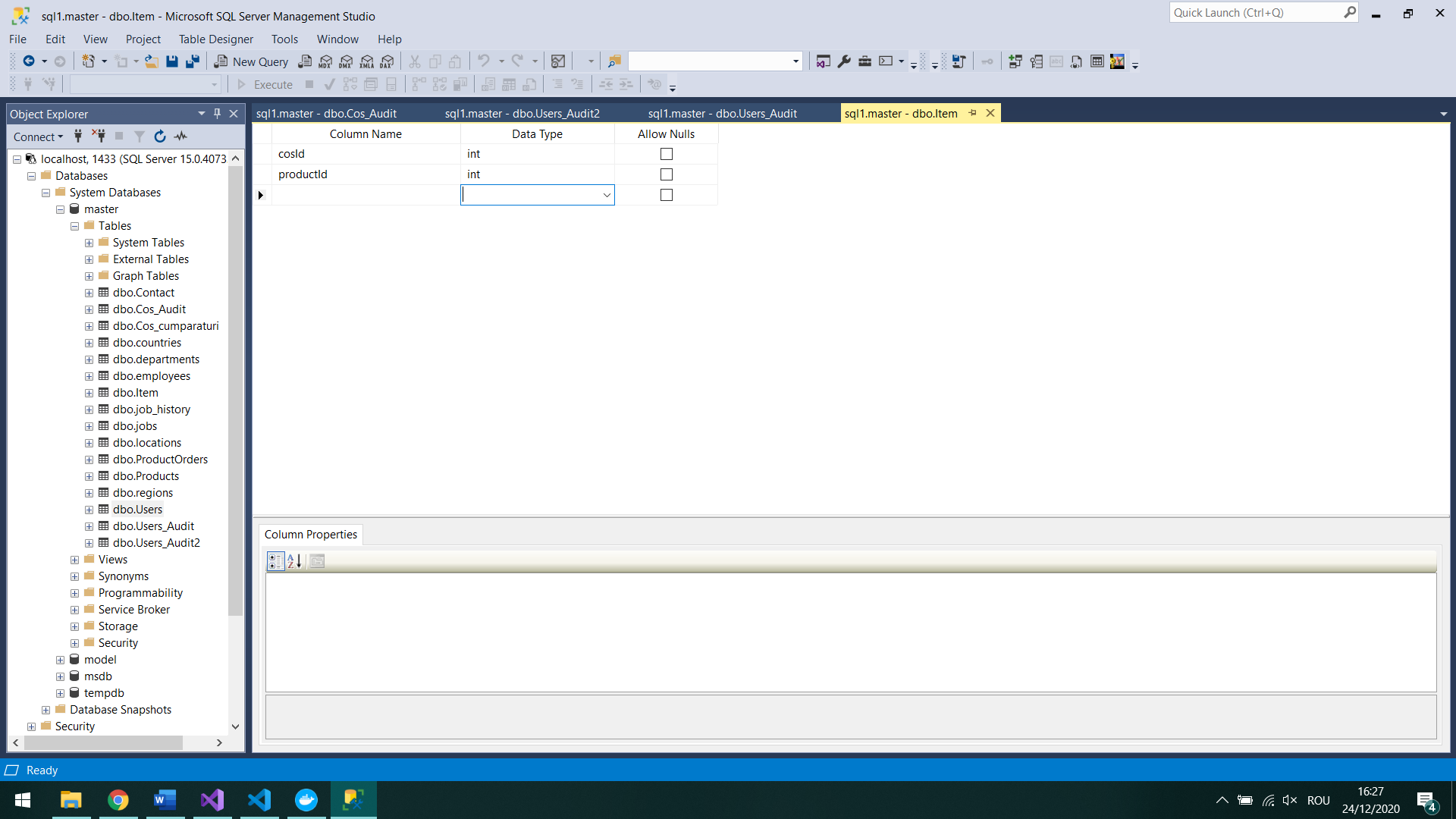
Figură ‑: Tabel Contact

1. Cos\_cumparaturi



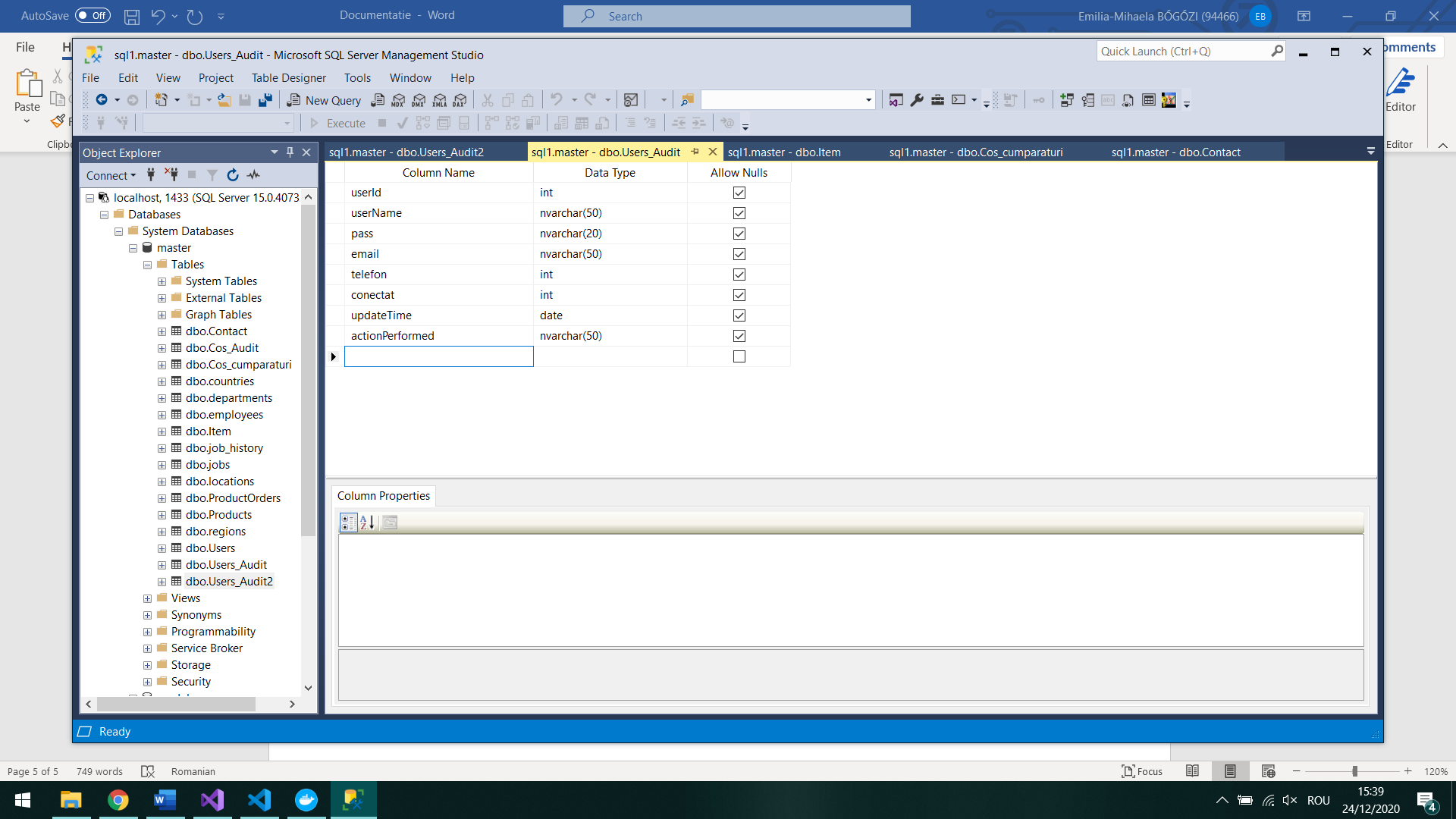
Figură ‑: Tabel Cos\_cumpărături

1. Item



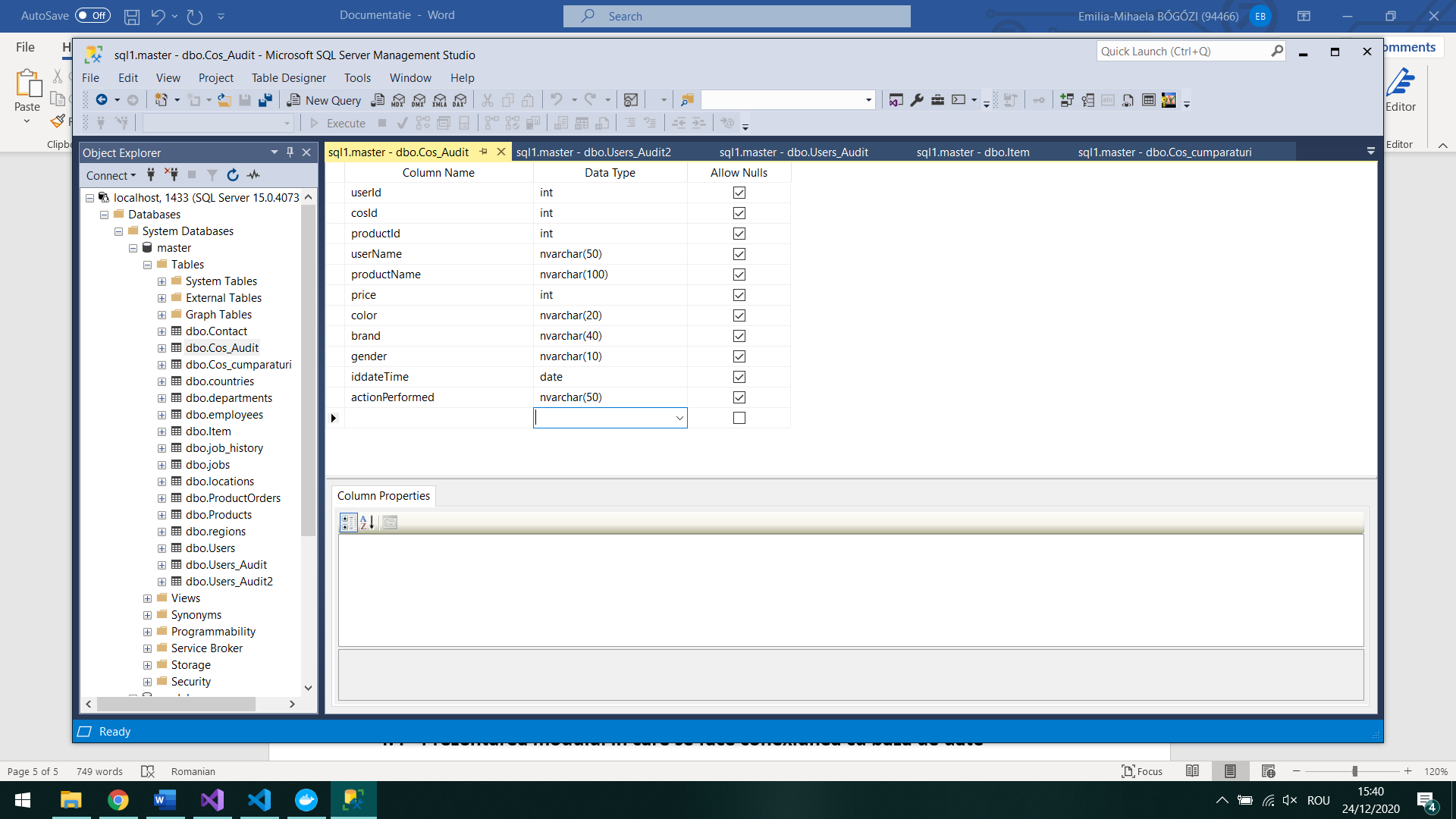
Figură ‑: Tabel Item

1. Users\_Audit



Figură ‑: Tabel Users\_Audit

1. Cos\_Audit



Figură ‑: Tabel Cos\_Audit

## Descrierea constrângerilor de integritate

1. NOT NULL

* În mod implicit, coloanele pot conține valori NULL. O constrângere NOT NULL în SQL este utilizată pentru a împiedica inserarea valorilor NULL în coloana specificată, considerându-l ca o valoare neacceptată pentru acea coloană. Aceasta înseamnă că ar trebui să furnizați o valoare SQL NOT NULL validă pentru acea coloană din instrucțiunile INSERT sau UPDATE, deoarece coloana va conține întotdeauna date. [2]
* Am folosit constrângerea NOT NULL în cadrul tabelelor:
* Users: userName, pass
* Products: productName, price, gender
* Contact: nume, prenume, email, mesaj
* Cos\_cumparaturi: userId, odate
* Item: cosId, productId

1. UNIQUE

* Acestă constrângere este utilizată pentru a se asigura că nu vor fi inserate valori duplicate într-o coloană specifică sau într-o combinație de coloane care participă la constrângerea UNIQUE și nu fac parte din CHEIA PRIMARĂ. [2]
* Indexul care este creat automat atunci când definiți o constrângere UNIQUE va garanta că nu există două rânduri în acel tabel care să aibă aceeași valoare pentru coloanele care participă la acel index, cu posibilitatea de a insera o singură valoare NULL unică în aceste coloane, dacă coloana permite NULL. [2]
* Am folosit această constrângere în cadrul definirii tabelei Users, pentru coloana username, deoarece nu vrem să existe 2 sau mai mulți utilizatori care să folosească același nume pentru conectarea în aplicație.

1. PRIMARY KEY

* Constrângerea PRIMARY KEY constă dintr-o coloană sau mai multe coloane cu valori care identifică în mod unic fiecare rând din tabel. [2]
* Constrângerea SQL PRIMARY KEY se combină între constrângerile UNIQUE și SQL NOT NULL, unde coloana sau setul de coloane care participă la cheia primară nu pot accepta o valoare NULL. Dacă cheia primară este definită în mai multe coloane, se put insera valori duplicate pe fiecare coloană în mod individual, dar valorile combinației tuturor coloanelor cheie primară trebuie să fie unice. [2]
* Constrângerea PRIMARY KEY este utilizată în principal pentru a impune integritatea entității tabelului. Integritatea entității asigură faptul că fiecare rând din tabel este o entitate identificabilă în mod unic. [2]
* Am folosit PRIMARY KEY în cadrul tabelelor:
* Users: userId
* Products: productId
* Cos\_cumparaturi: cosId

1. FOREIGN KEY

* O cheie străină este o cheie de bază de date care este utilizată pentru a lega două tabele împreună. Constrângerea FOREIGN KEY identifică relațiile dintre tabelele bazei de date referindu-se la o coloană sau un set de coloane, în tabelul Child care conține cheia externă, la coloana PRIMARY KEY sau set de coloane, în tabelul părinte. [2]
* Am folosit FOREIGN KEY în cadrul tabelelor:
* Cos\_cumparaturi: userId
* Item: cosId, productId

## Descrierea procedurilor și funcțiilor

1. PROCEDURE select\_products

* Procedura selectează toate produsele din tabela Products.

1. PROCEDURE new\_contact

* Primește ca parametru: nume, prenume, email, telefon, mesaj;
* Procedura inserează în tabelul Contact datele primite ca parametru.

1. PROCEDURE create\_new\_account

* Primește ca parametru: userName, password, email, telefon;
* Parametrii funcției reprezintă datele completate la crearea unui cont nou;
* Procedura verifică dacă există un utilizator cu numele specificat ca parametru în tabelul Users, iar dacă nu există inserează utilizatorul în tabel.

1. PROCEDURE login\_user

* Primește ca parametru: userName, password;
* Procedura selectează username-ul și parola din tabela Users care corespund cu cele specificate ca parametru.

1. PROCEDURE connect\_user

* Primește ca parametru userName;
* Procedura actualizează tabelul Users astfel: caută în tabelul Users utilizatorul cu id-ul specificat ca parametru și îi modifică valoarea „conectat” în 1, adică utilizatorul este conectat.

1. FUNCTION get\_user\_id

* Primește ca parametru userName și returnează un număr INT;
* Funcția returnează id-ul userului cu numele specificat ca parametru.

1. PROCEDURE add\_cos\_cumparaturi
   * Primește ca parametru userId;
   * Procedura verifică dacă există un utilizator cu id-ul specificat ca parametru în tabelul Cos\_cumparaturi; dacă nu există, îl adaugă.
2. PROCEDURE get\_products\_by\_user

* Primește ca parametru userName;
* Procedura selectează toate produsele din coșul de cumpărături ai utilizatorului cu numele specificat ca parametru.

1. FUNCTION sum\_products

* Primește ca parametru userName și returnează un număr INT;
* Funcția returnează suma elementelor din coșul de cumpărături al unui utilizator identificat prin numele specificat ca parametru.

1. FUNCTION count\_products

* Primeste ca parametru userName si returneaza un numar INT;
* Functia returneaza cate elemente are in cosul de cumparaturi un utilizator identificat prin numele trimis ca parametru.

1. PROCEDURE insert\_item

* Primește ca parametru userName, productId;
* Procedura inserează în tabelul Item coșul corespunzator utilizatorului cu numele specificat ca parametru și id-ul produsului;

1. PROCEDURE delete\_item

* Primește ca parametru: userName, productId;
* Procedura elimină prima apariție a produsului cu id-ul specificat ca parametru din coșul de cumpărături (tabelul Item) al utilizatorului cu username-ul specificat ca parametru.

1. PROCEDURE delete\_all\_items

* Primește ca parametru userName;
* Procedura este utilizată pentru eliminarea tuturor produselor din tabelul Cos\_cumparaturi al unui utilizator specificat ca parametru în cazul finalizării unei comenzi.

1. PROCEDURE disconnect\_user

* Primește ca parametru userName;
* Procedura actualizează tabelul Users astfel: caută în tabelul Users utilizatorul cu id-ul specificat ca parametru și îi modifică valoarea „conectat” în 0, adică utilizatorul este deconectat.

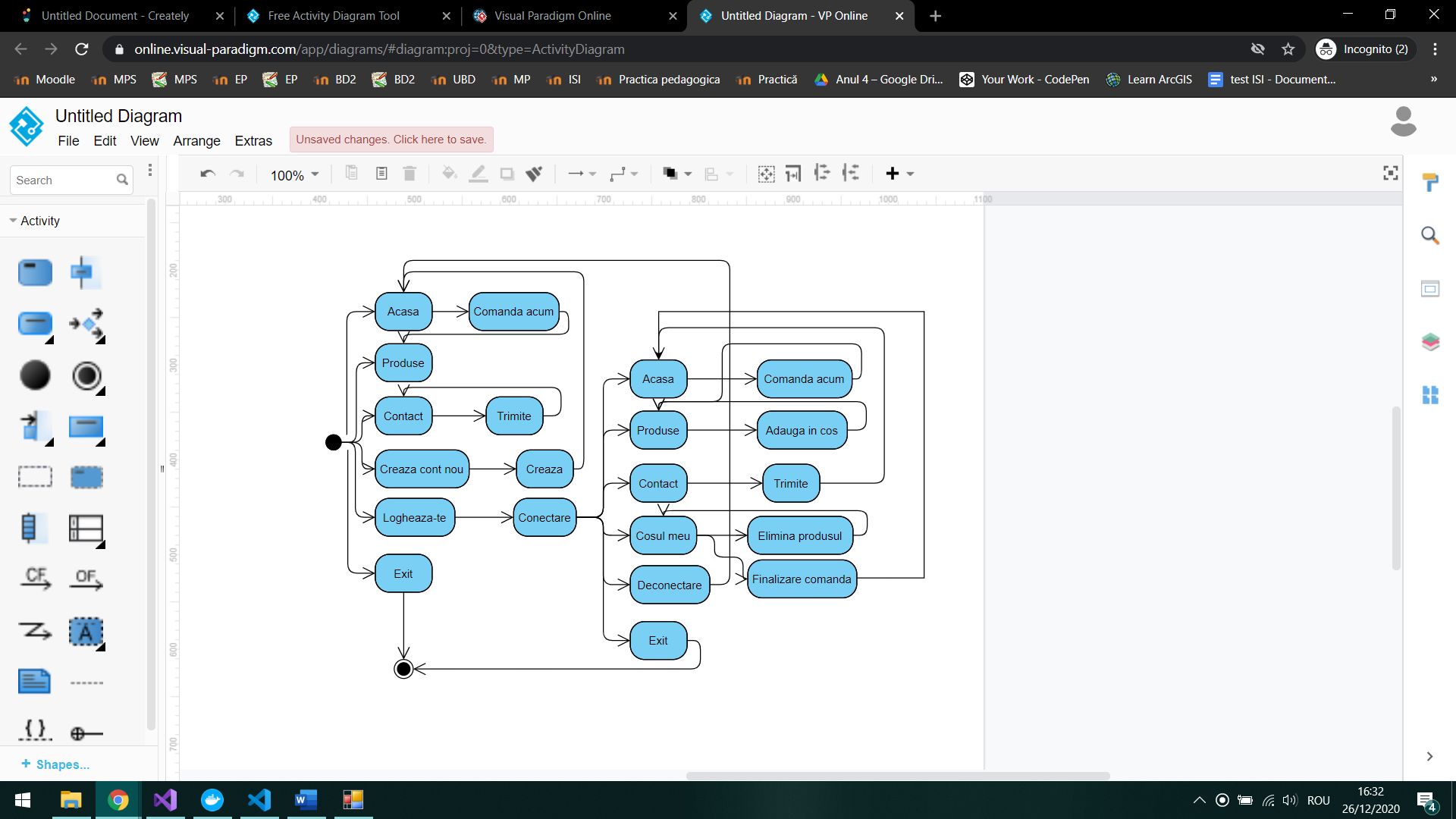
# Descrierea aplicației

## Diagrama și structura claselor

Graphical user interface, application

Description automatically generated

## Diagrama de stări și fluxul de lucru (workflow) pentru aplicație



Figură ‑: Diagrama de stări

## Prezentarea modului în care se face conexiunea cu baza de date

Am instalat un container MS SQL Server cu urmatoarele specificații:

* Numele bazei de date: localhost
* Port: 1433
* Username: SA
* Parola: parolaAiaPuternic4!

În programul C# am realizat conexiunea astfel:

|  |
| --- |
| SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();  builder.DataSource = "localhost, 1433";  builder.UserID = "SA";  builder.Password = "parolaAiaPuternic4!";  builder.InitialCatalog = "";  SqlConnection connection = new SqlConnection(builder.ConnectionString);  connection.Open();  ...  Connection.Close(); |

# Capturi de ecran pentru interfețe și rapoarte

|  | NECONECTAT | CONECTAT |
| --- | --- | --- |
| Home |  |  |
| Produse |  |  |
| Contact |  |  |
| Coșul meu | - |  |
| Crează cont nou |  | - |
| Loghează-te |  | - |

# Concluzii

Proiectul intitulat „Magazin online de ceasuri” a fost finalizat cu succes. Sistemul a fost dezvoltat cu multă grijă și fără erori și, în același timp este eficient și consumă puțin timp. Scopul acestui proiect a fost dezvoltarea unei aplicații care folosește o bază de date SQL Server pentru achiziționarea de articole dintr-un magazin.

Acest proiect ne-a ajutat să obținem informații valoroase și cunoștințe practice pe mai multe subiecte, cum ar fi proiectarea unei aplicatii Desktop folosind C#, utilizarea tool-urilor puse la dispoziție de acestea și gestionarea bazei de date folosind SQL Server.

# Bibliografie

|  |  |
| --- | --- |
| [1] | S. R. Vatrapu, „Design and Implementation of E-Commerce Site for Online Shopping,” 2014. |
| [2] | A. Yaseen, „SQLShack,” 25 October 2017. [Interactiv]. Available: https://www.sqlshack.com/commonly-used-sql-server-constraints-not-null-unique-primary-key/. |