Liceul de Informatică “Tiberiu Popoviciu”, Cluj-Napoca

**PROIECT PENTRU OBȚINEREA ATESTĂRII**

**PROFESIONALE ÎN INFORMATICĂ**

Titlul lucrări:

**Gestiunea unor restaurante:**

**-Klausen-Burger**

**-Rao**

**Cluj-Napoca Trif Marius Andrei**

**Mai 2023 Clasa a XII-a C**

**Coordonatori: prof: Lazar Laura**

**prof: Eugen Claudiu Persa**

**Cuprins:**

1. **Prezentarea generală a temei**
2. **Motivarea alegerii subiectului**
3. **Resurse software necesare**
4. **Descrierea lucrări**
5. **Concluzii**
6. **Bibliografie**

**1.Prezentarea generală a temei**

Această aplicație realizează gestionarea unor baze de date ale restaurantelor: Klausen-Burger, Rao. Această baza de date prezintă atât locația fiecărui restaurant dar totodata si stocurile de produse pe care fiecare restaurant în parte le are, furnizorii cu care cooperează restaurantele, meniurile împarțite pe categorii si preparatele care se află în fiecare categorie dar totodată si compoziția fiecărui aliment.

Această bază de date stochează informații despre:

* Restaurant
* Stocuri
* Furnizori
* Meniuri
* Preparate
* Compoziția preparatelor

Aplicația gestionează cantitățiile de alimente pe care un restaurant le are pe stoc dar si cantitățiile de alimente care se folosesc pentru fiecare preparat în parte, permițând modificarea acestora după dorințele manager-ului restaurantului.

**2. Motivarea alegerii subiectului**

Am ales acest subiect deoarece am dorit să îmi pun în joc imaginația astfel să pot dezvolta o nouă interfață a unei aplicații sau viitoare aplicații, care să permită unor manageri sau patroni de restaurante folosirea ei astfel putând gestiona mult mai ușor stocurile de alimente pe care aceștia le consumă/le folosesc pe parcursul unor săptămâni sau chiar luni la restaurantul lor. Această aplicație are scopul de a eficientiza achizițiile de alimente in cadrul firemei, ele putând fi aciziționate în momentul în care pe stoc se împuținează astfel reducând la maxim aruncarea acestora din cauza expirarii sau depașiri termenului de valabilitate, dar si o mai bună gestionare a datelor privind productivitatea dar si consumul diferitor alimente din cadrul restaurantului. Acest lucru ajută la o mai bună aproximare pentru momentul achiziției diferitor tipuri de aliment.

**3. Resurse software necesare**

Realizarea programului s-a făcut cu ajutorul aplicației Microsoft Visual Studio 2010. Aceasta asigurând diferite facilități:

* Crearea unei baze de date, incluzând atât tabelele, legăturile cât și diferite interogări
* Popularea tabelelor
* Modificarea unor date deja existente în baza de date
* Extragerea diferitor informatii din tabel
* Asigurarea unui meniu privind folosirea acestuia

**4. Descrierea lucrări**

**4.1 ERD**

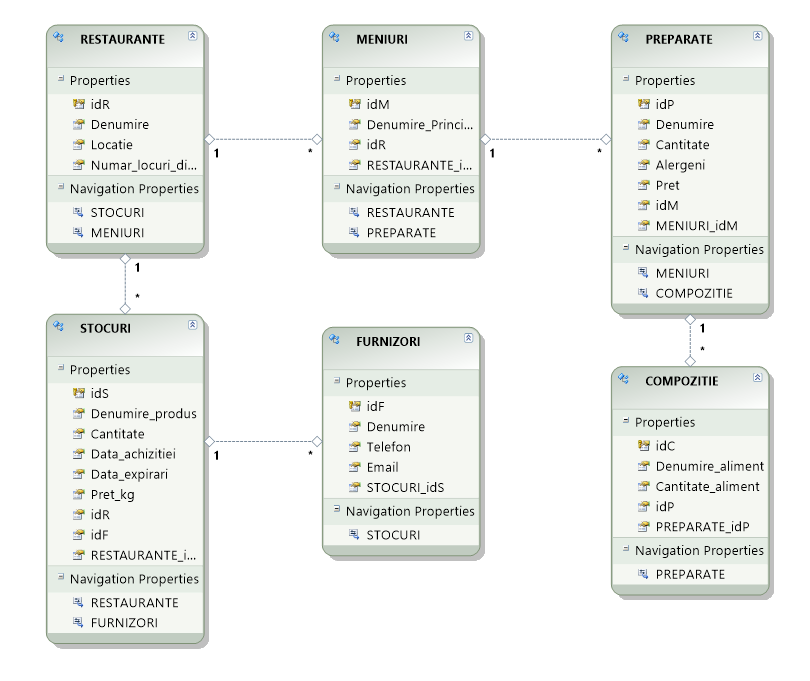


Fig.1

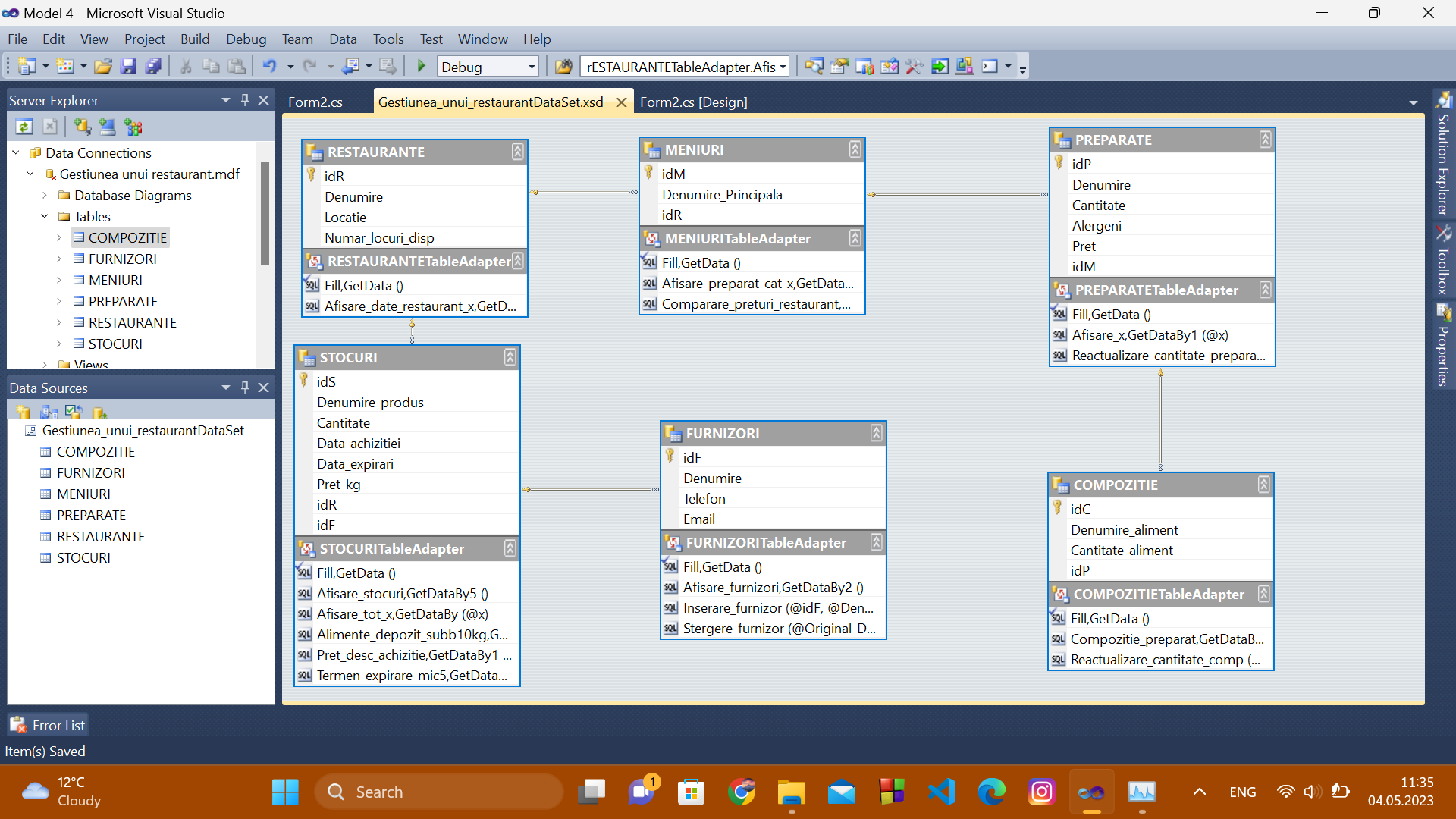


Fig.2

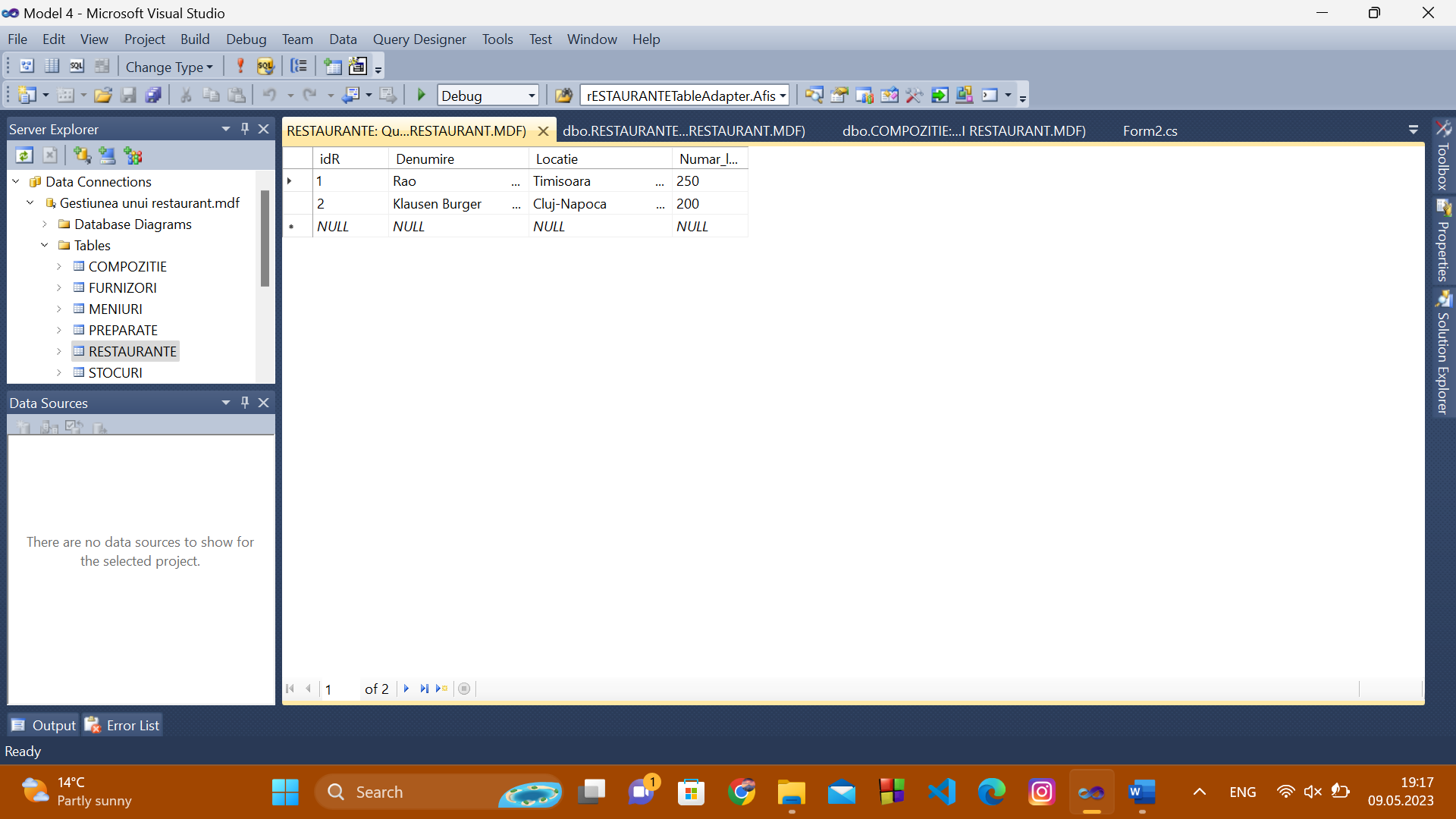
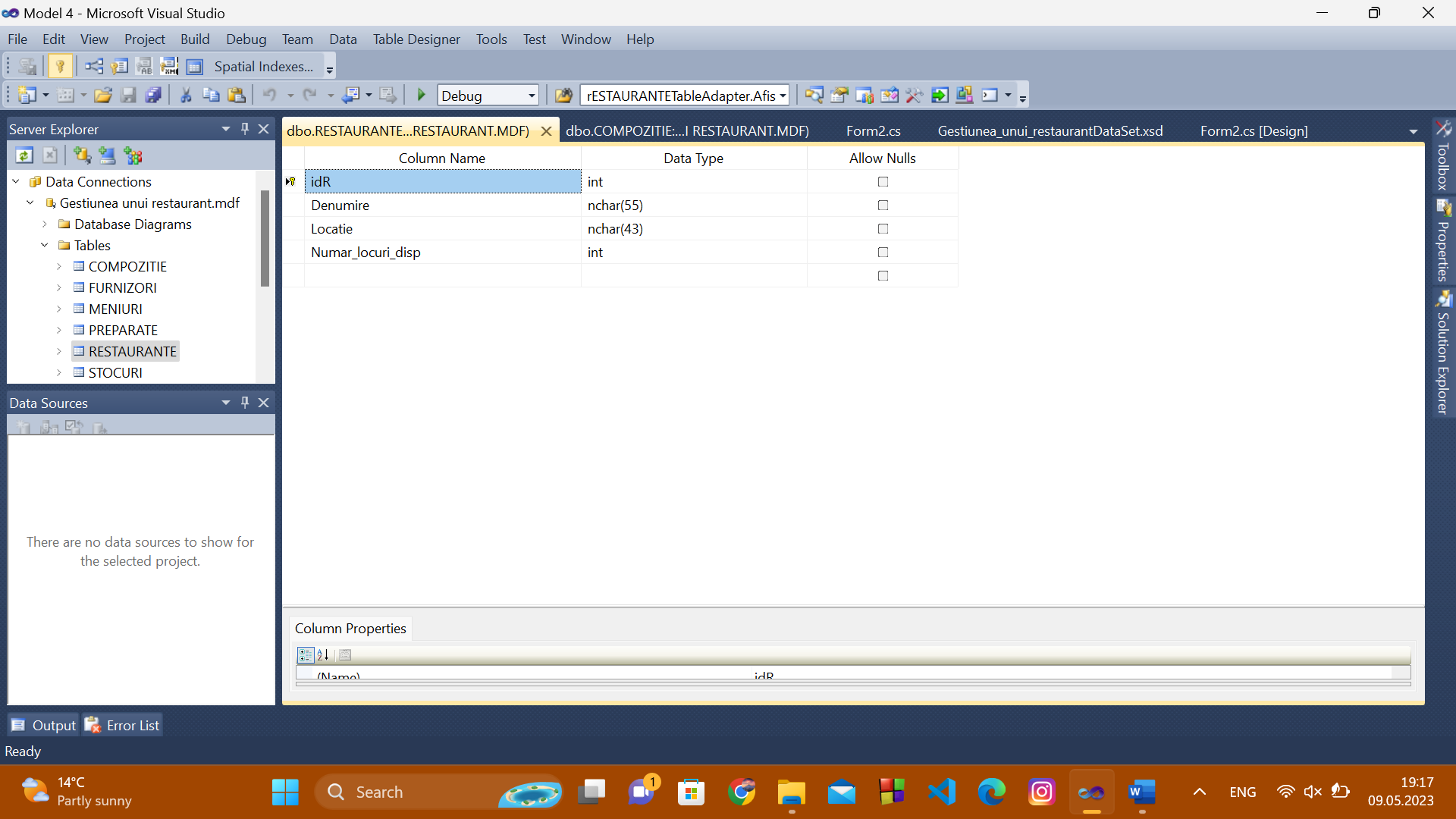
Pe baza primei diagrame se poate observa diferite legături între tabele, astfel reiese faptul că:

1. Un restaurant are unul sau mai multe stocuri ale produselor pe care acesta le folosește în incinta localului.
2. Stocurile pot fi aprovizionate de unul sau mai multi furnizori, totul depinzând de managerul restaurantului si planului de bussines dar si de contractele pe care acesta le are.
3. Totodată un restaurant poate avea unul sau mai multe meniuri în funcție de sezon, de traficul de persoane dar și de feedback-ul pe care restaurantul îl are.
4. Meniul trebuie să aibă unul sau mai multe preparate, din care putem afla denumirea acestora, cantitatea, prețul, dacă conține alergeni sau nu si categoria din care acesta face parte(Ex. Aperitiv, Supe, băuturi nonalcolice, etc.).
5. Fiecare preparat în parte este compus din unul sau mai multe alimente. Astfel în tabelul compoziție putem afla compoziția fiecărui aliment in parte, putând și modifica gramajul acestuia.

**4.2 Structural**

Din punct de vedere structural, aplicația este configurată din 6 tabele interconectate între ele.

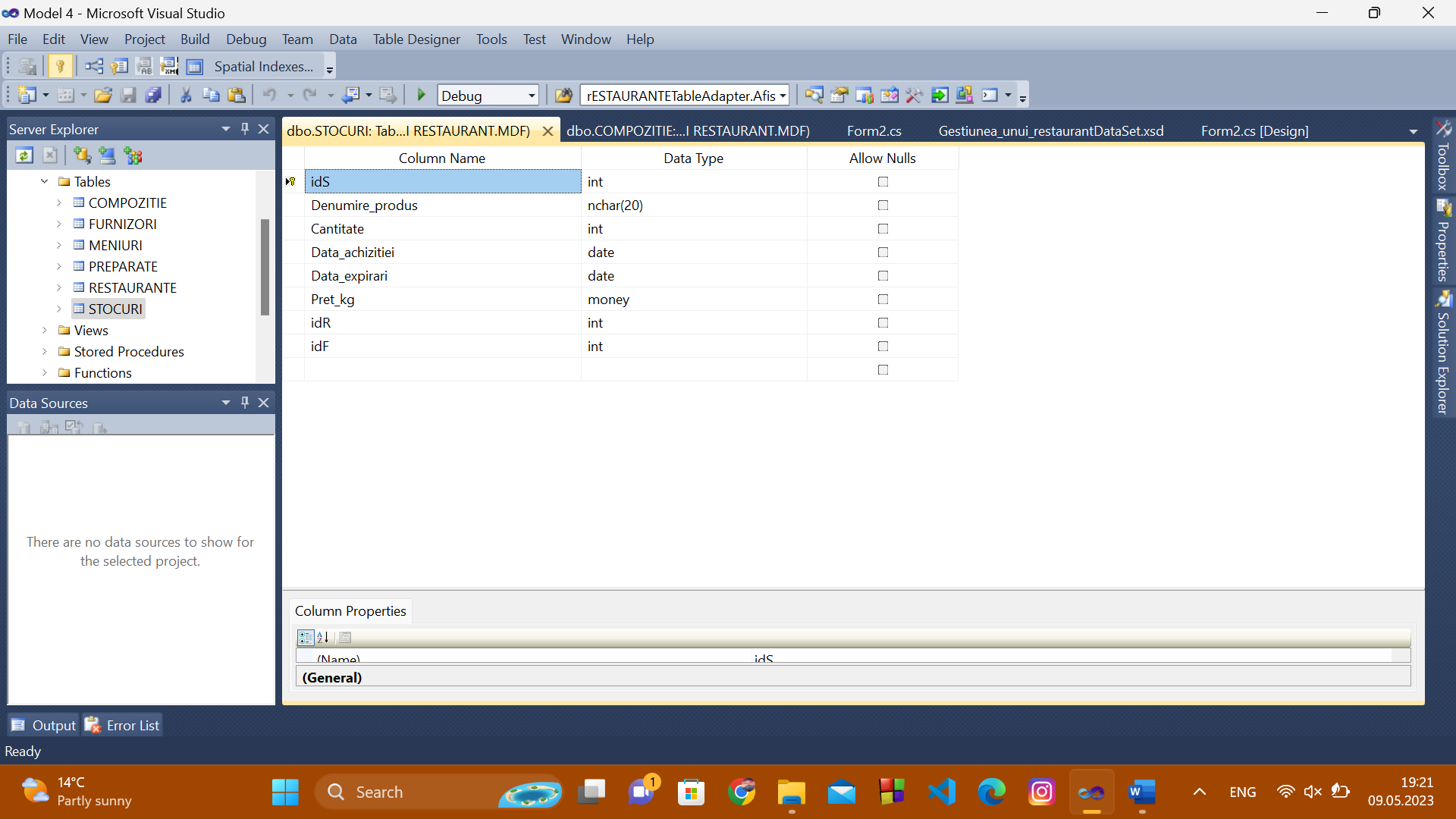
**Tabel RESTAURANTE**

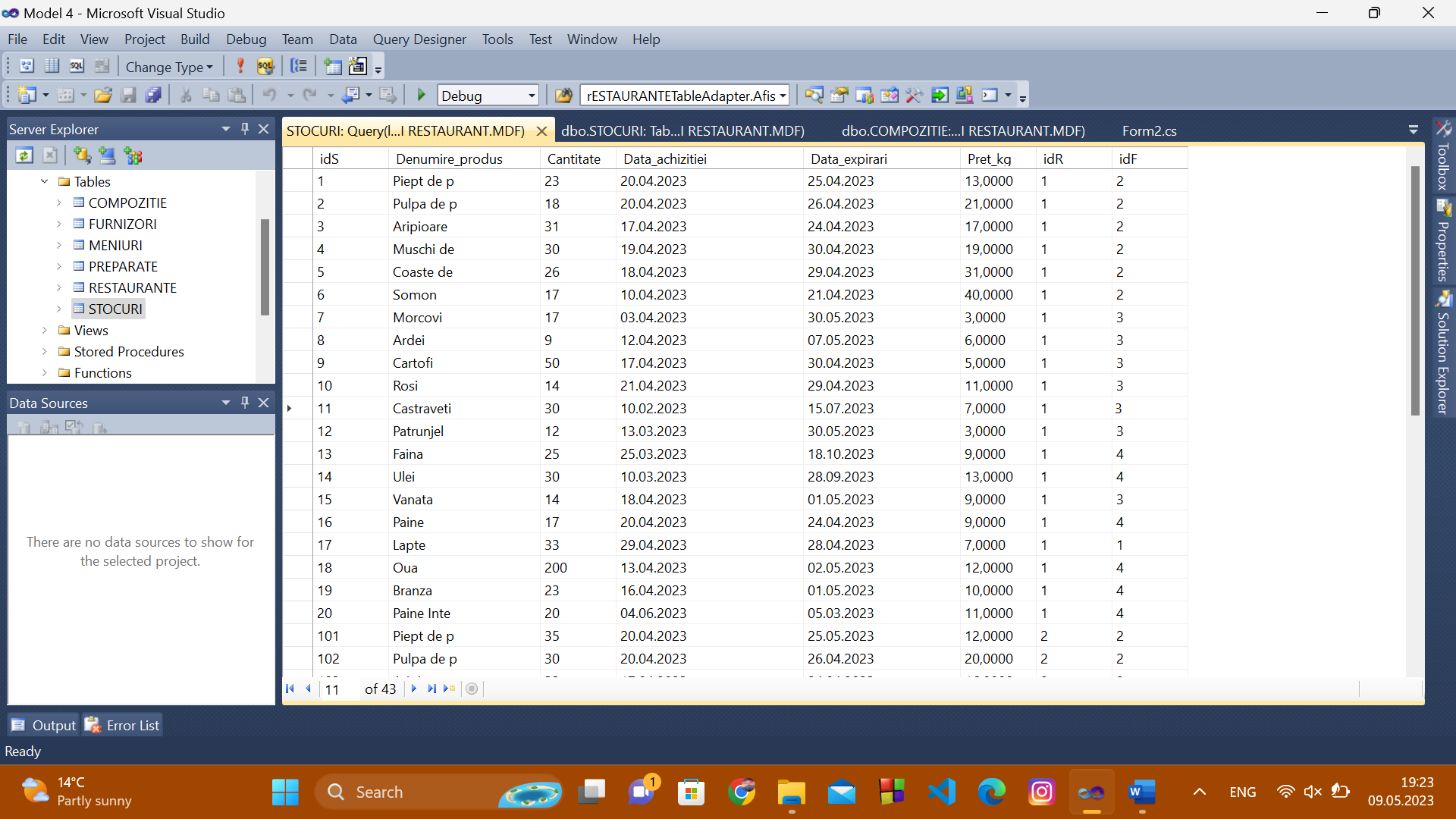


Tabelul Restaurante conține următoarele informații despre fiecare local in parte:

* Un id specific fiecărui restaurant
* O denumire a restaurantului
* Locația acestuia
* Numărul de locuri total de care acesta dispune

**Tabel STOCURI**

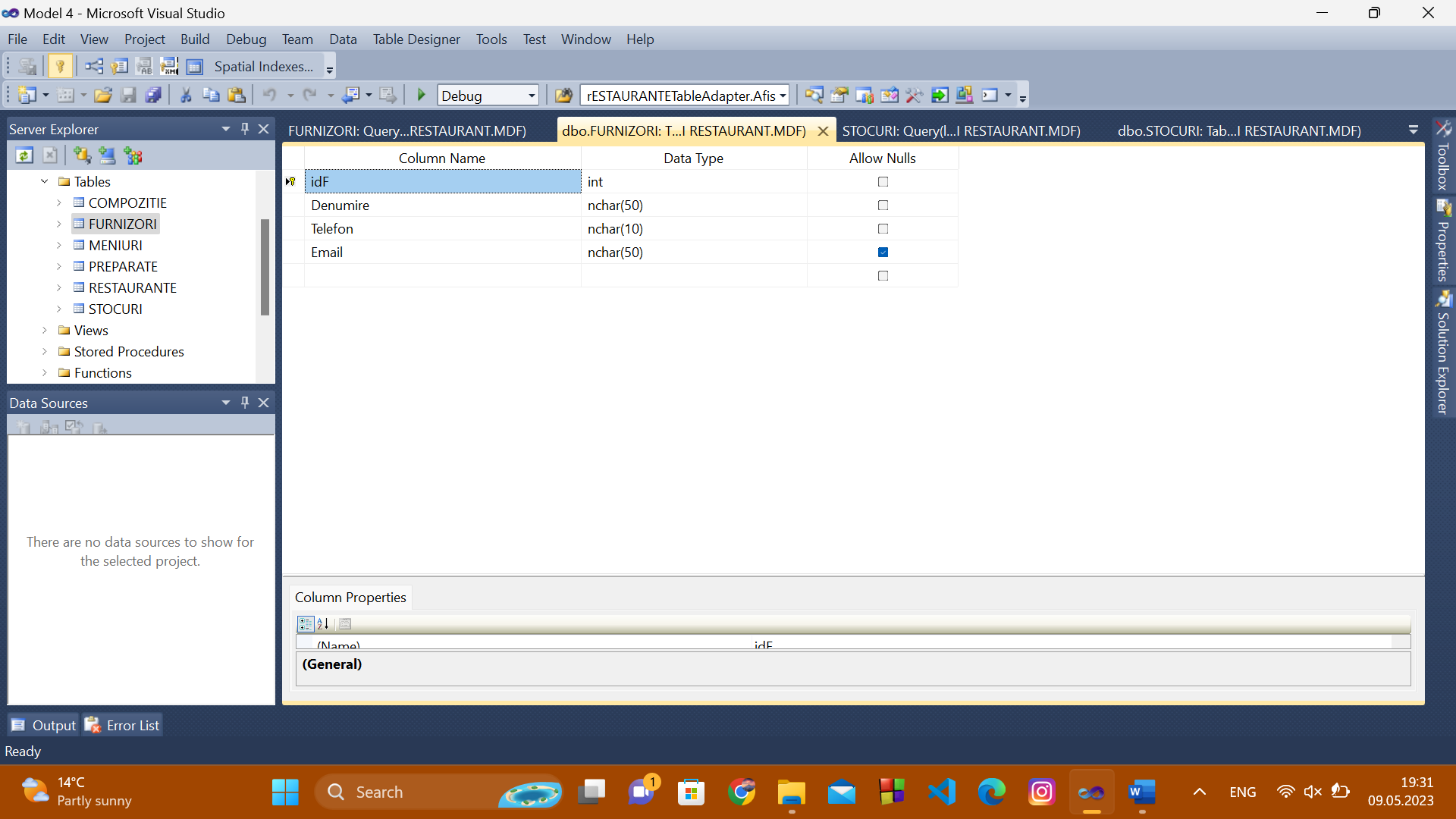


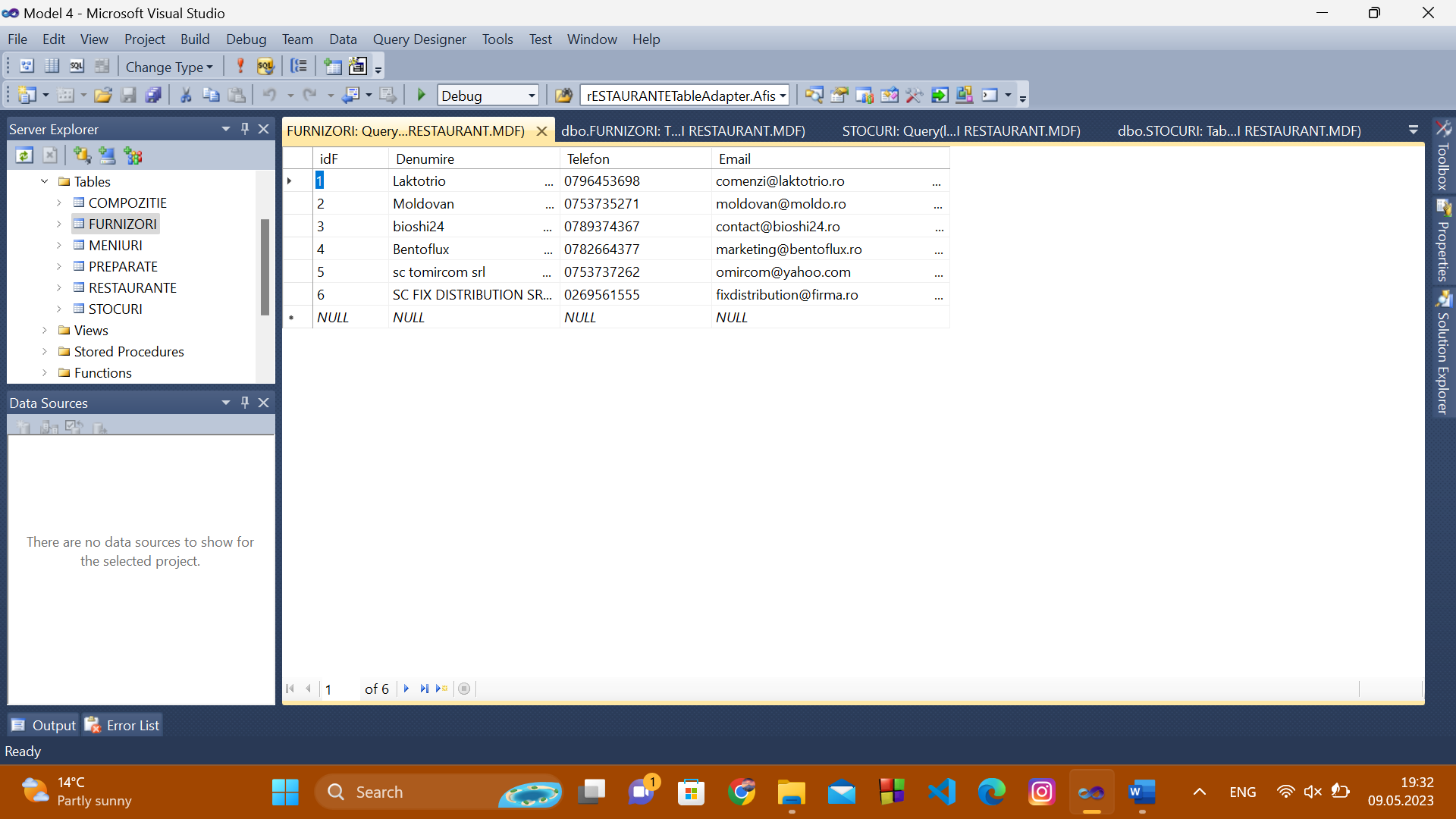


Tabelul Stocuri conține următoarele informații despre cantitățile de produse pe care fiecare restaurant le are in gestiunea sa până in momentul folosiri:

* Un id specific fiecărui produs in parte
* Denumirea produsului
* Cantitatea care se află pe stocul localului
* Data achiziției mărfii
* Data expirării
* Pretul achiziției per kg
* O cheie străină idR prin care pot lega unul sau mai multe stocuri de restaurantul pe care îl doresc
* O cheie străină idF prin care pot lega unul sau mai mulți furnizori de stocurile pe care le am

**Tabel FURNIZORI**

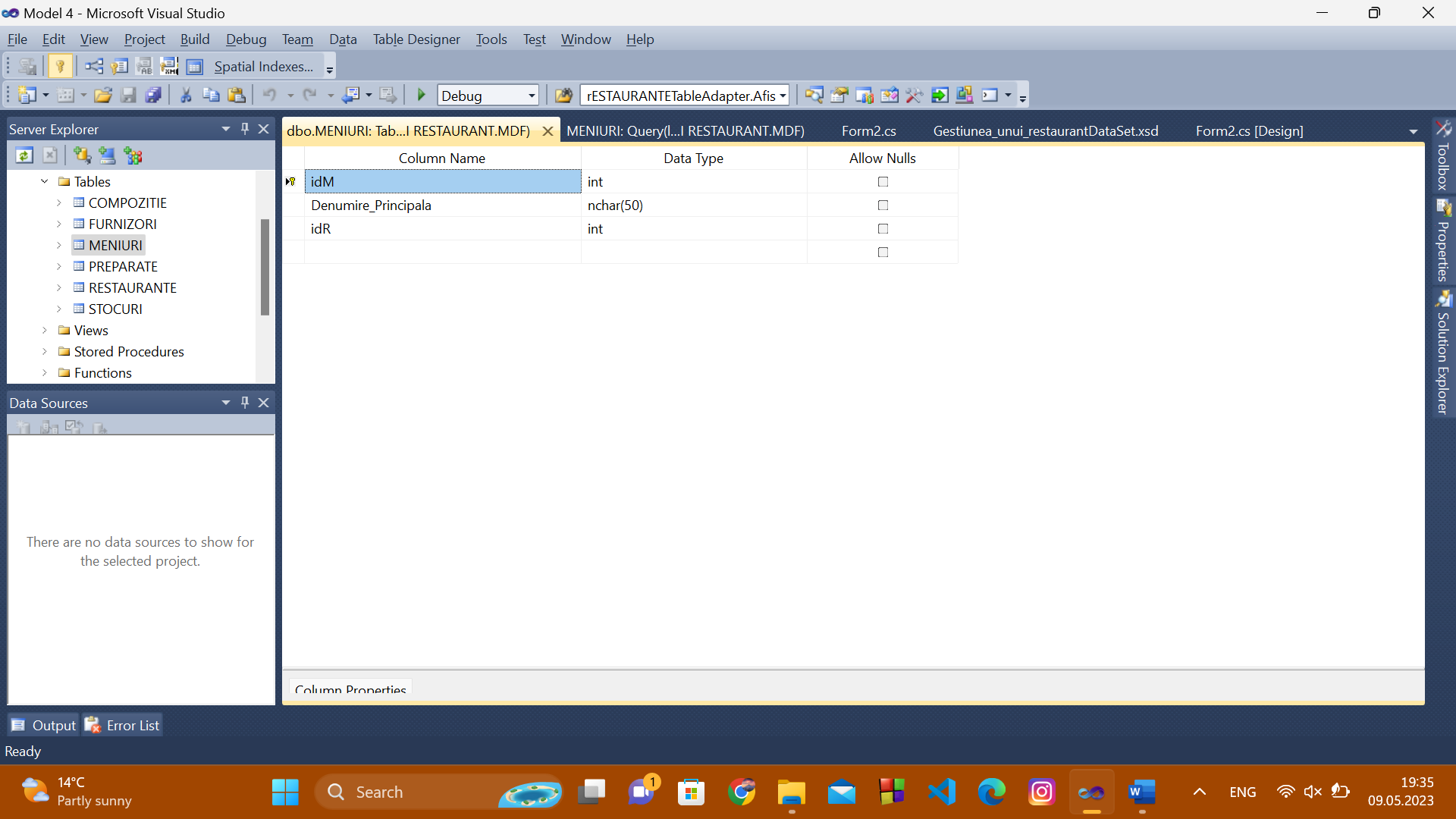


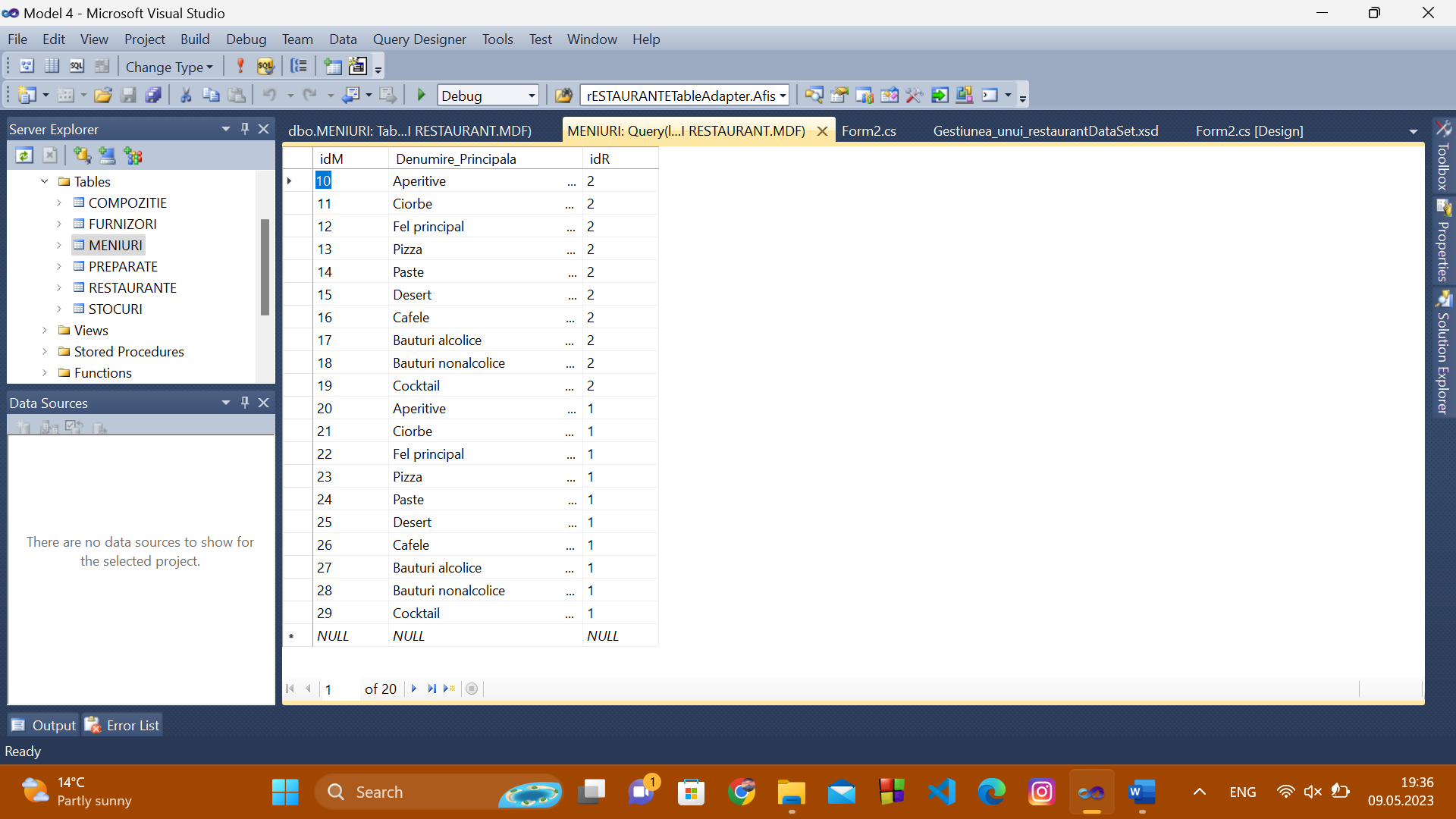


Tabelul Furnizori conține următoarele informații despre fiecare furnizor in parte:

* Un id specific fiecărui furnizor in parte
* Denumirea firmei
* Numărul de telefon al firmei
* Email

**Tabel MENIURI**

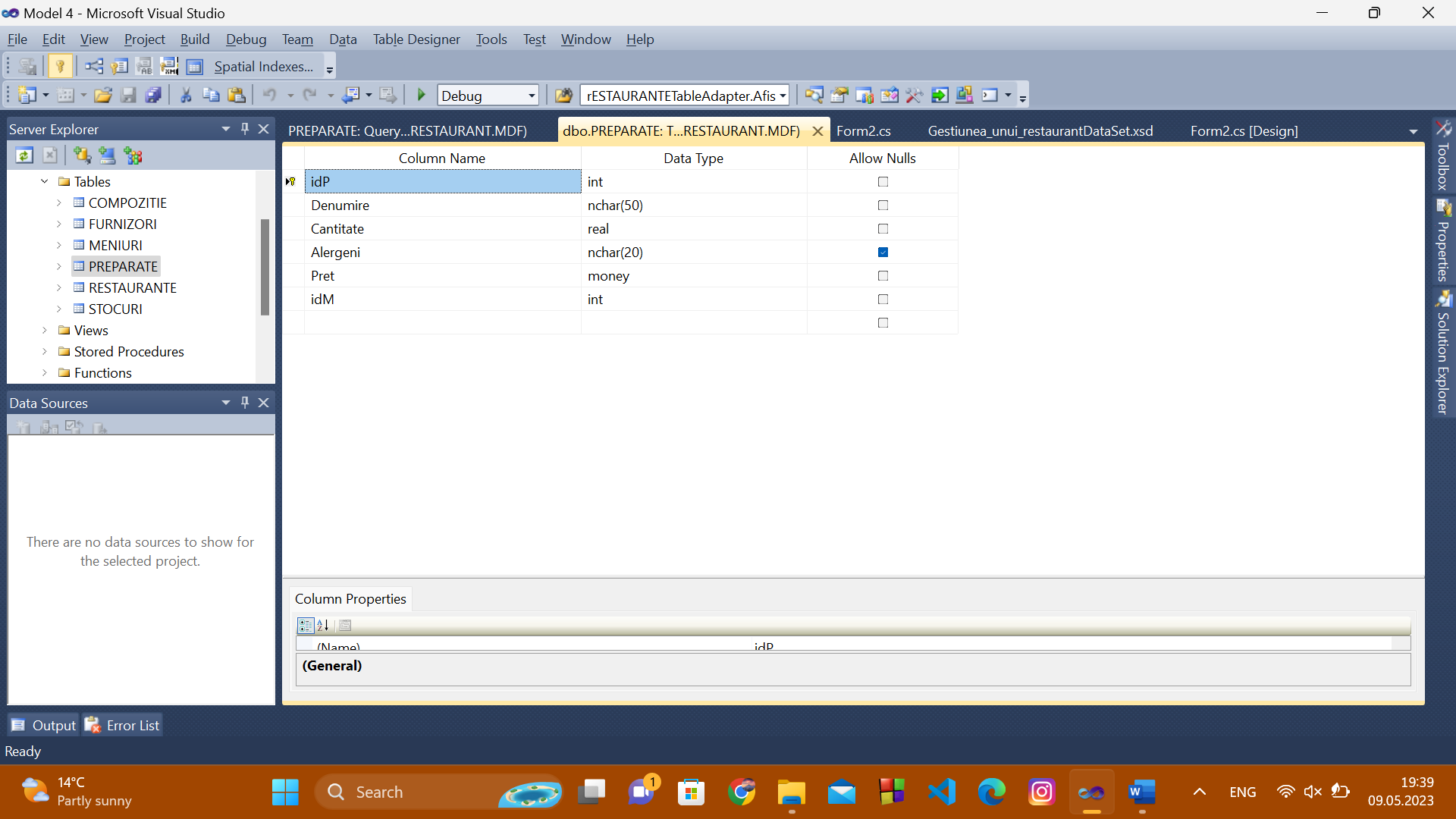


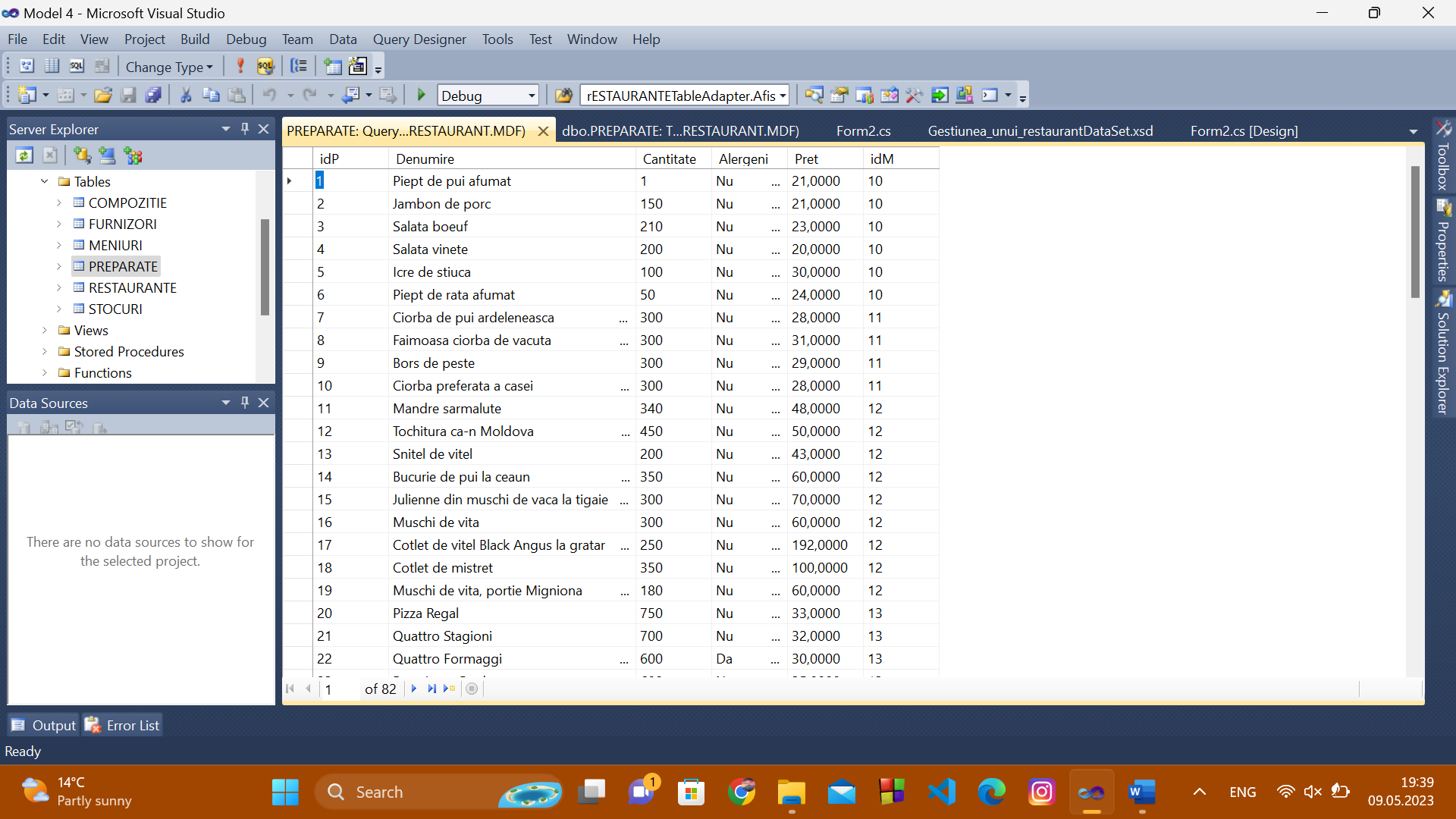


Tabelul Meniuri conține urmatoarele informați despre fiecare meniu in parte:

* Id specific fiecărui meniu
* Denumirea principală a capitolului de mâncare
* O cheie străină idR prin care pot lega unul sau mai mule categorii din meniul pe care îl am de un restaurant

**Tabel PREPARATE**

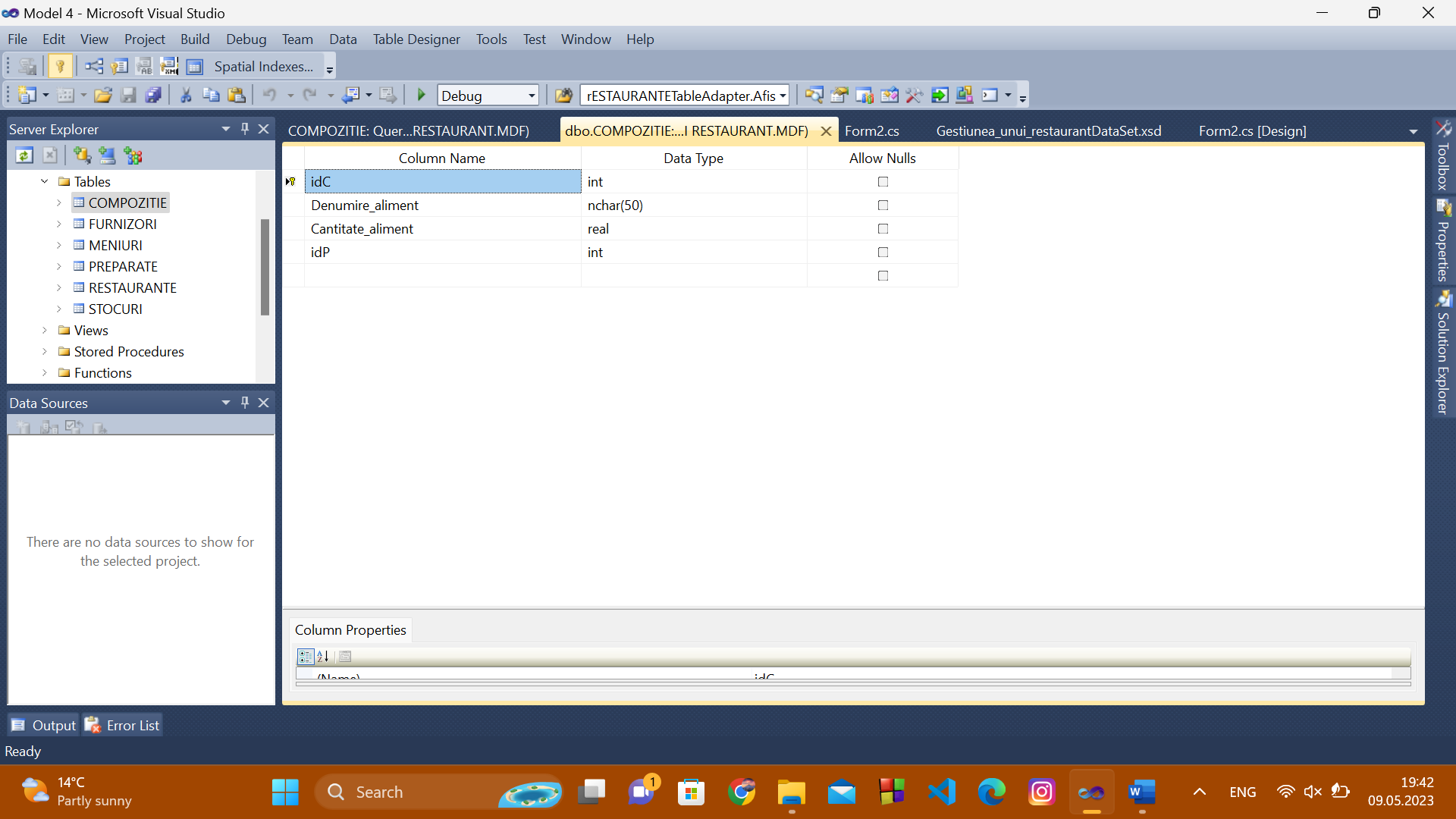


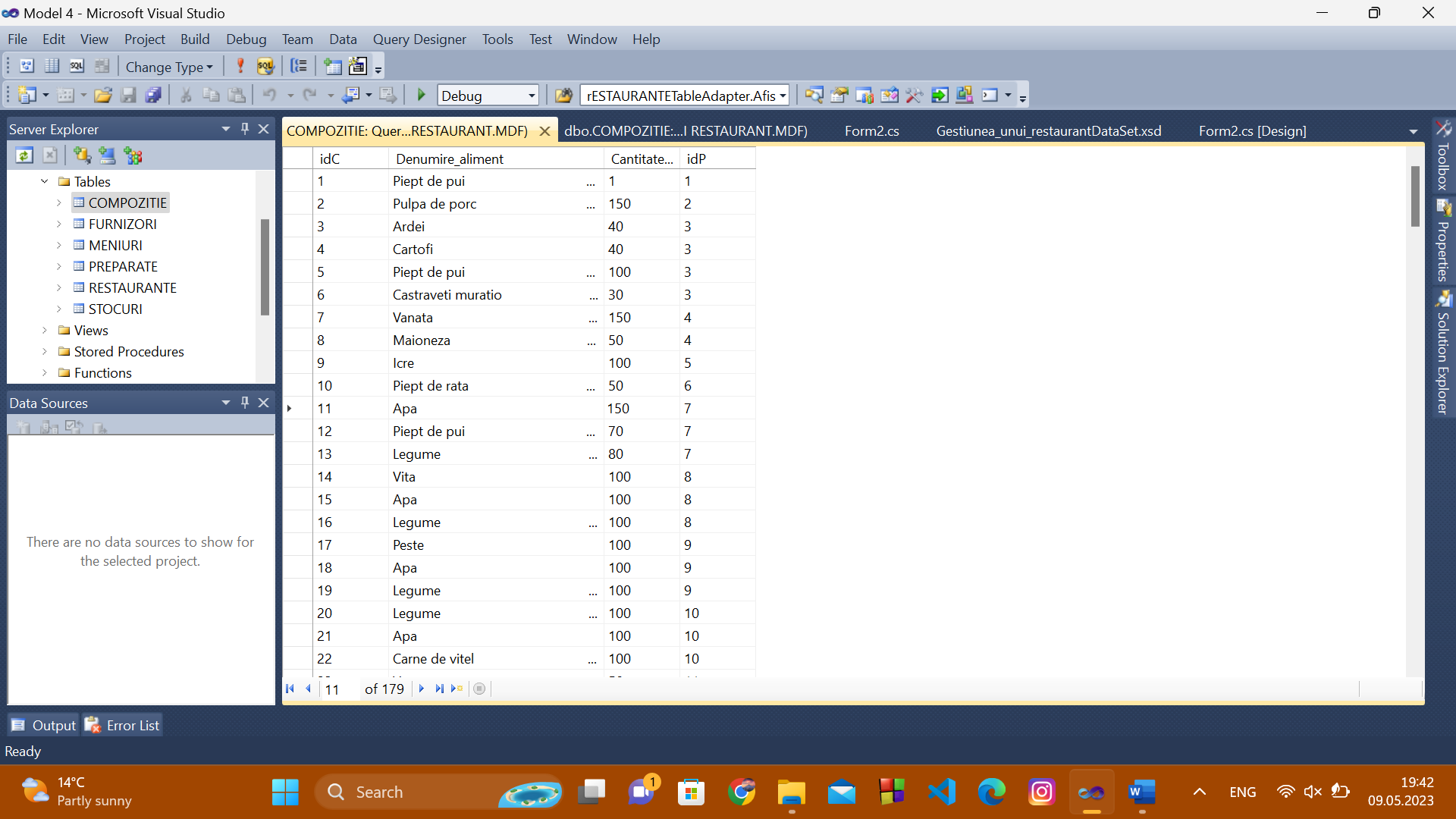


Tabelul Preparate conține urmatoarele informații despre fiecare preparat in parte:

* Id specific fiecărui preparat
* Denumirea principală a preparatului
* Cantitatea
* Daca are alergeni sau nu
* Prețul acestuia
* O cheie străina idM prin care pot lega unul sau mai mule preparate de categoria din meniu din care acesta face parte

**Tabel PREPARATE**

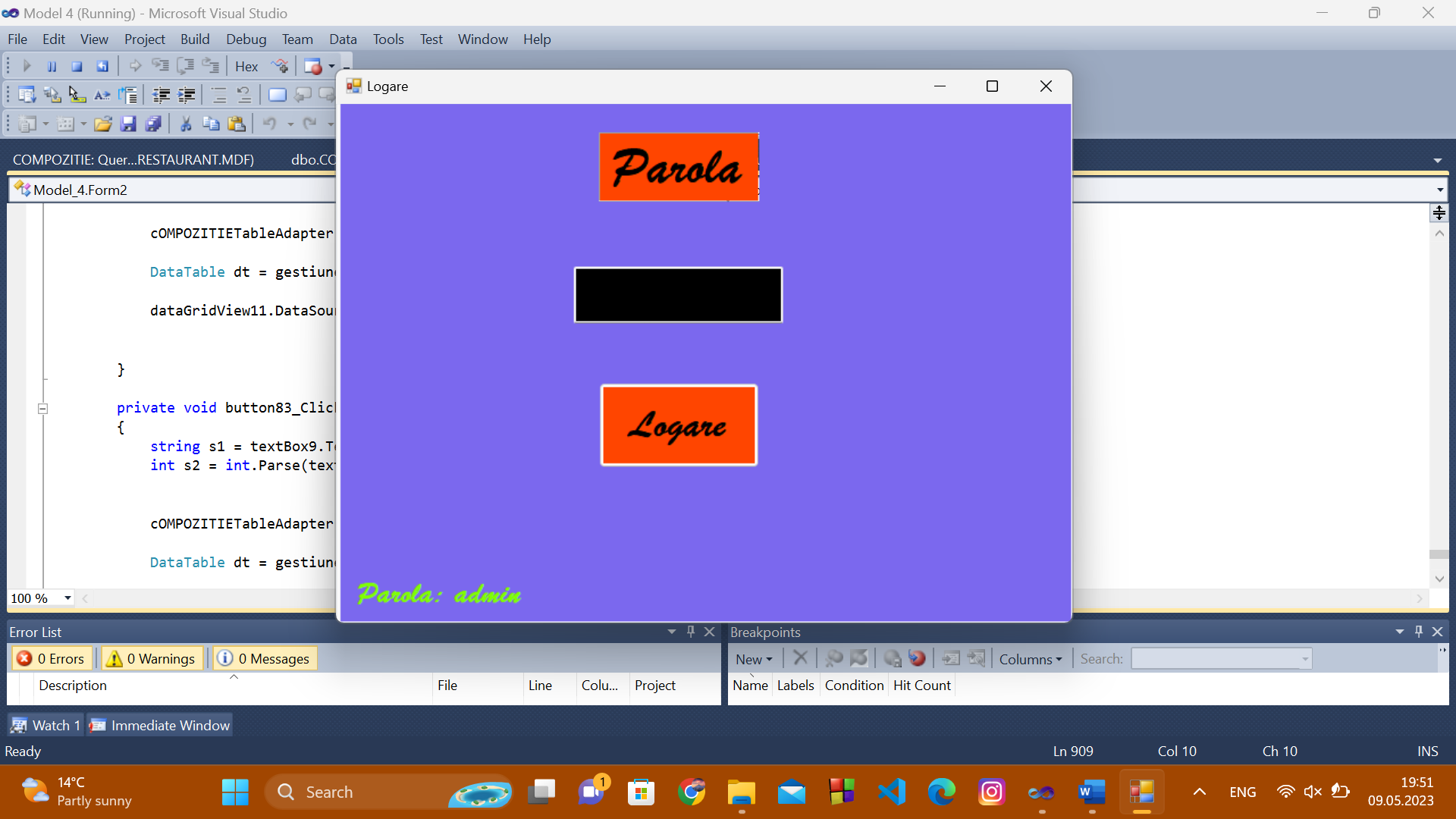




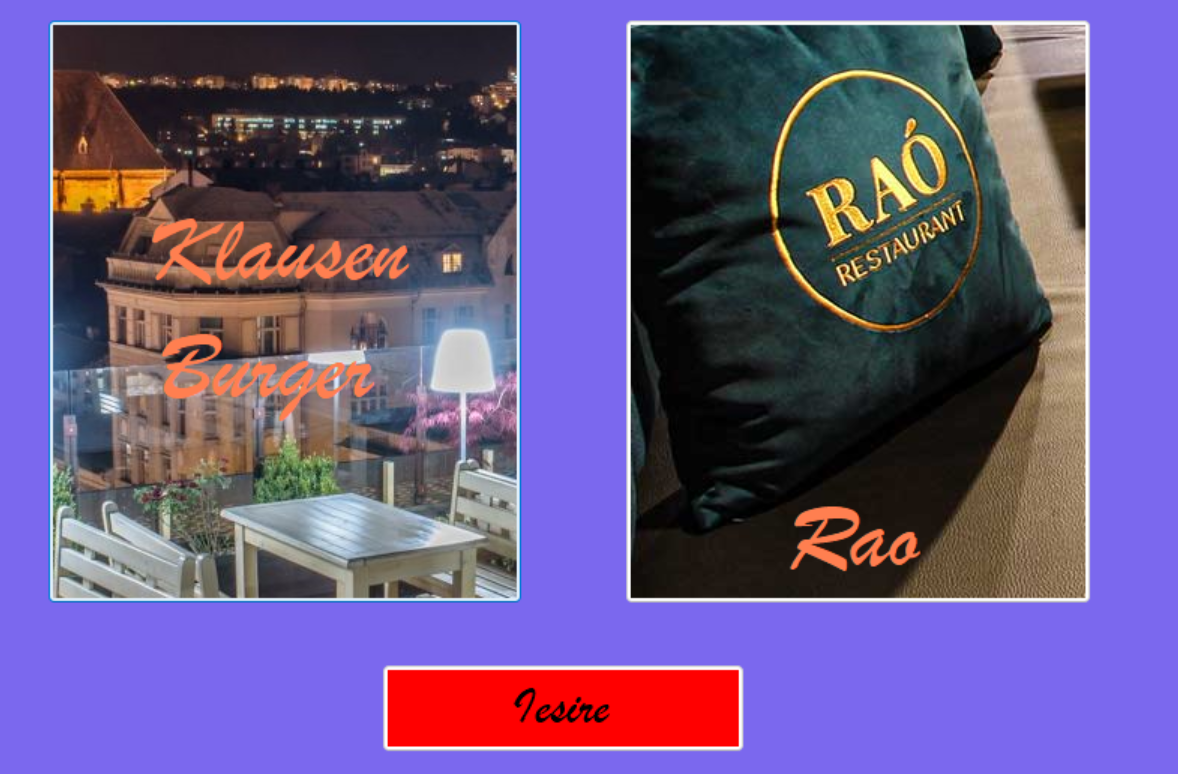
Tabelul Compoziție conține urmatoarele informații despre fiecare preparate in parte, si din ce este acesta compus:

* Id specific fiecărui aliment care compune un preparat
* Denumirea alimentului care compune preparatul
* Cantitatea
* O cheie straină idP prin care pot lega unul sau mai mule alimente de un preparat

**Pagina de pornire a aplicației**



In prim plan se poate observa faptul că la rularea programului ne este cerută o parolă, acest lucru ajutând utilizatorul la o mai bună protectie a datelor dar totodată si daunele materiale care s-ar putea produce dacă aceasta nu ar fi.



Mai apoi trebuie să ne selectăm restaurantul în care dorim să vedem diferite date sau să le modificăm.



În această sectiune ne este prezentat un tabel în care avem în prim plan cerinta de a afisa produsele care expiră în mai putin de 5 zile

Interogare:

SELECT Data\_achizitiei, Data\_expirari, Denumire\_produs, Pret\_kg, idF, idR, idS, Cantitate

FROM STOCURI

WHERE (DATEDIFF(day, Data\_achizitiei, Data\_expirari) < 5) AND (idR = @x)

idR fiind selectat automat de la alegerea restaurantului pe care dorim să îl vizionăm/ să lucrăm în baza de date a acestuia.

Codul de implementare:

private void button14\_Click(object sender, EventArgs e)

{

tabControl1.SelectedTab = tabPage6;

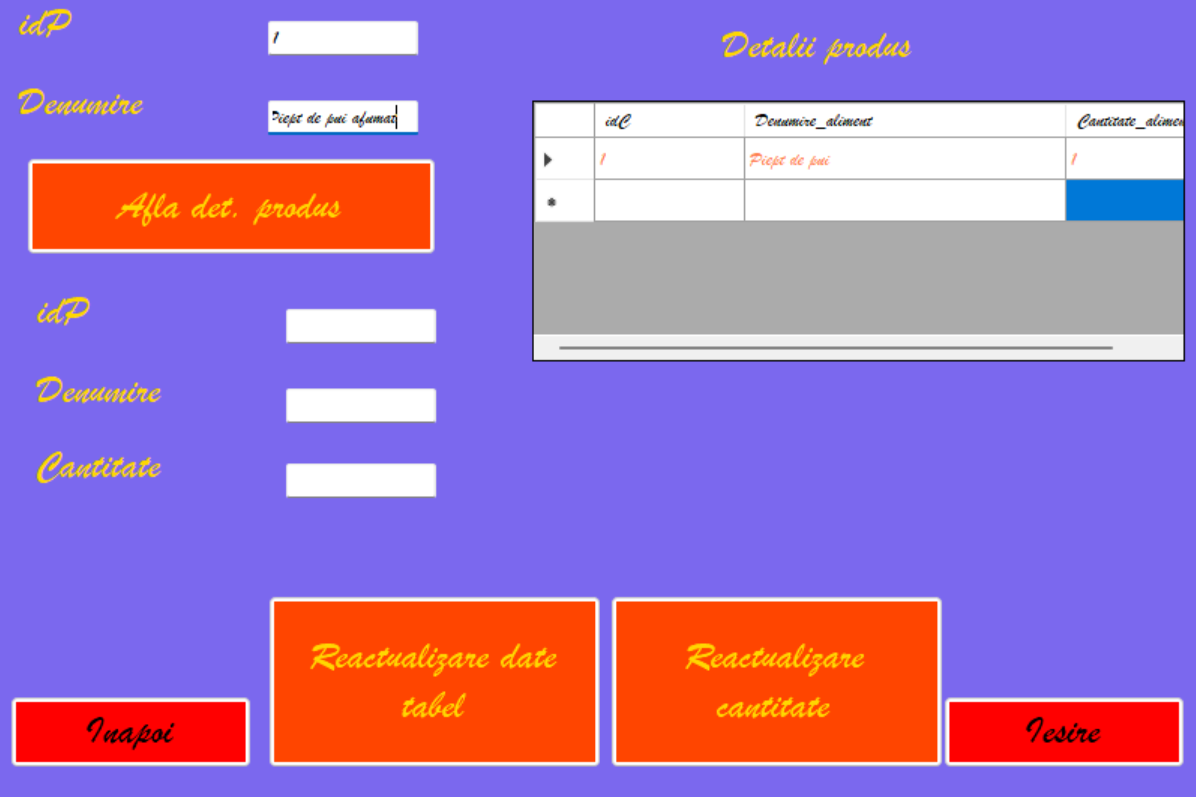
int x = 1;

sTOCURITableAdapter.Termen\_expirare\_mic5(gestiunea\_unui\_restaurantDataSet.STOCURI,x);

DataTable dt = gestiunea\_unui\_restaurantDataSet.STOCURI;

dataGridView3.DataSource = dt;

}



Interogare:

-detalii produs

SELECT COMPOZITIE.Denumire\_aliment, COMPOZITIE.Cantitate\_aliment, COMPOZITIE.idC

FROM COMPOZITIE INNER JOIN

PREPARATE ON COMPOZITIE.idP = PREPARATE.idP

WHERE (PREPARATE.Denumire = @y) AND (PREPARATE.idP= @x)

-reactualizare cantitate

UPDATE COMPOZITIE

SET Cantitate\_aliment = @Cantitate\_aliment

WHERE (Denumire\_aliment = @Original\_Denumire\_aliment) AND (idC = @Original\_idC);

Cod implementare:

-detalii produs

private void button83\_Click(object sender, EventArgs e)

{

string s1 = textBox9.Text;

int s2 = int.Parse(textBox8.Text);

cOMPOZITIETableAdapter.Compozitie\_preparat(gestiunea\_unui\_restaurantDataSet.COMPOZITIE, s1, s2);

DataTable dt = gestiunea\_unui\_restaurantDataSet.COMPOZITIE;

dataGridView12.DataSource = dt;

}

-reactualizare date

private void button39\_Click(object sender, EventArgs e)

{

int s1 = int.Parse(textBox25.Text);

string s2 = textBox24.Text;

float s = float.Parse(textBox18.Text);

cOMPOZITIETableAdapter.Reactualizare\_cantitate\_comp(s, s2, s1);

string s3 = textBox9.Text;

int s4 = int.Parse(textBox8.Text);

cOMPOZITIETableAdapter.Compozitie\_preparat(gestiunea\_unui\_restaurantDataSet.COMPOZITIE, s3, s4);

DataTable dt1 = gestiunea\_unui\_restaurantDataSet.COMPOZITIE;

dataGridView11.DataSource = dt1;

textBox25.Clear();

textBox24.Clear();

textBox18.Clear();

}

**5.Concluzii**

Aceasta aplicație ar putea fi folosita chiar și în viață reala daca aceasta ar primi diferite update-uri pentru a putea fi folosita măcar pe un telefon mobil, dar totodată și prezenta a mai multor interogări de inserare sau de ștergere datelor în momentul în care acestea sunt folosite. Mai poate fi valorificata aceasta aplicație prin implementarea unei alarme pentru produsele care mai au câteva zile până la expirare. Aplicația putând fi înțeleasă cu ușurință și totodată și folosita sau învățată într-un timp foarte scurt.

**6.Bibliografie**

Aceasta aplicație a fost inspirate din lipsa de informații asupra unor baze de date cu ajutorul cărora se poate gestiona un restaurant, dar și din dorință de a putea eficientiza diferite procese prin care poate unii manageri/ patroni de restaurante pot căștiga mai mult timp prin folosirea ei.