Weronika Chamera

PROJEKT

„Książka Kucharska”

Temat: Baza przechowuje informacje dotyczące przepisów kulinarnych, które dodawane są przez użytkowników.

Zaprojektowana baza stanowi zbiór przepisów kulinarnych. Składa się z ośmiu tabel powiązanych ze sobą relacjami. Służy ona ułatwieniu dostępu do informacji na temat przygotowywanej potrawy. Wśród nich możemy odnaleźć dane dotyczące poszczególnego przepisu, listy składników, właściwości potrawy czy nawet osoby, która jest jej autorem. Baza w sposób stosunkowo łatwy i szybki pozwala zapoznać się z zamieszczonymi danymi, a także usprawnia proces wyszukiwania interesujących nas zagadnień. Dzięki temu określone kryteria pozwalają automatycznie wygenerować użytkownikom zindywidualizowane propozycje.

Lista Tabel:

- **Przepisy**
  Tabela zawiera informacje dotyczące wprowadzanego przepisu (m.in. nazwę przepisu, czas jego przygotowania czy datę jego wpisu do bazy)

- **Produkty_spozywcze**
  Posiada informacje o produktach dodanych do bazy, ich średnie koszty i kalorie wyrażone jednostkowo co do ustalonej dla produktu miary.

- **Miara**
  Zawiera zbiór jednostek miary w jakich wyrażane są produkty spożywcze używane w przepisach.

- **Rodzaj_kuchni**
  Zawiera zbiór rodzajów kuchni, które kategoryzują przepisy według ich pochodzenia – jak np. kuchnia włoska.

- **Skladniki**
  Tabela ta łączy w sobie dane dotyczące przepisu z potrzebnymi do jego realizacji produktami tworząc listę składników.

- **Autorzy**
  Posiada informacje na temat autorów przepisów – ich dane osobowe.

- **Typ_dania**
  Zawiera zbór typów dań do jakich możemy przypisać dany przepis – jak np. śniadanie czy obiad.

- **Poziom_trudności**
  Zawiera dane określające poziom trudności przepisu w oparciu o ilość dopuszczalnych składników.

DEFINICJE TABEL

TABELA - Przepisy			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Przepis	int	IDENTITY	<PK>
Nazwa_przepisu	NVARCHAR (50)	UNIQUE, NOT NULL	<FK>
ID_ Au	int	NOT NULL	<FK>
ID_Poz_trudnosci	int	NOT NULL	<FK>
ID_Danie	int	NOT NULL	<FK>
ID_Kuchnia	int	NOT NULL	<FK>
Ilosc_porcji	int	NOT NULL	
Data_zapisu	datetime	NOT NULL	DEFAULT Getdate()
Czas_przygotowania	time(7)	NOT NULL	
Okładka	image		

TABELA - Produkty_spozywcze			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Produkt	int	IDENTITY	<PK>
Nazwa_produktu	NVARCHAR (50)	UNIQUE, NOT NULL	
Kcal_jednostkowe	FLOAT	NOT NULL	
Koszt_jednostkowy	FLOAT	NOT NULL	
ID_ Jed_miary	int	NOT NULL	<FK>

TABELA - Miara			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Miara	int	IDENTITY	<PK>
Jednostka_miary	VARCHAR (4)	UNIQUE, NOT NULL	

TABELA - Rodzaj_kuchni			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Rodzaj_Kuchni	int	IDENTITY	<PK>
Nazwa_kuchni	NVARCHAR (25)	UNIQUE, NOT NULL	

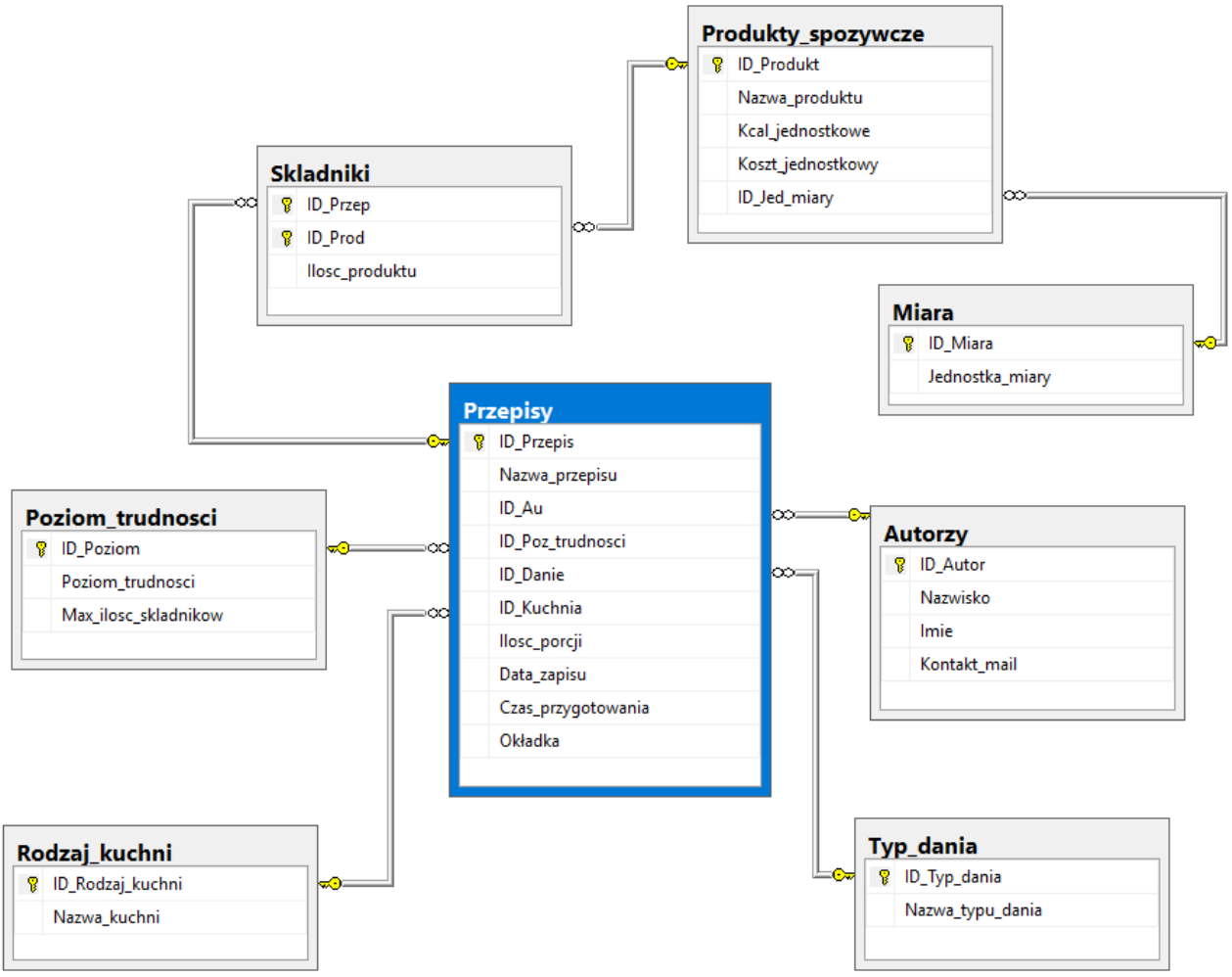
TABELA - Skladniki			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_ Przep	int	NOT NULL	<PK>
ID_Prod	int	NOT NULL	<PK>
Ilosc_produktu	int	NOT NULL	

TABELA - Autorzy			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Autor	int	IDENTITY	<PK>
Nazwisko	NVARCHAR (35)	NOT NULL	
Imie	NVARCHAR (25)	NOT NULL	
Kontakt_mail	NVARCHAR (45)		

TABELA - Typ_dania			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_Typ_dania	int	IDENTITY	<PK>
Nazwa_typu_dania	NVARCHAR (20)	UNIQUE, NOT NULL	

TABELA – Poziom_trudnosci			
Nazwa pola	**Typ danych**	**Właściwości**	**Uwagi**
ID_ Poziom	int	IDENTITY	<PK>
Poziom_trudnosci	NVARCHAR (15)	UNIQUE, NOT NULL	
Max_ilosc_skladnikow	int	NOT NULL	

DIAGRAM



Równania relacji:

dbo.Produkty_spozywcze.ID_ Jed_miary = dbo. Miara.ID_Miara

dbo.Produkty_spozywcze.ID_Produkt = dbo.Składniki.ID_Prod

dbo.Przepisy.ID_Przepis = dbo.Składniki.ID_ Przep

dbo.Przepisy.ID_Au = dbo.Autorzy.ID_ Autor

dbo.Przepisy.ID_Danie = dbo.Typ_dania.ID_ Typ_dania

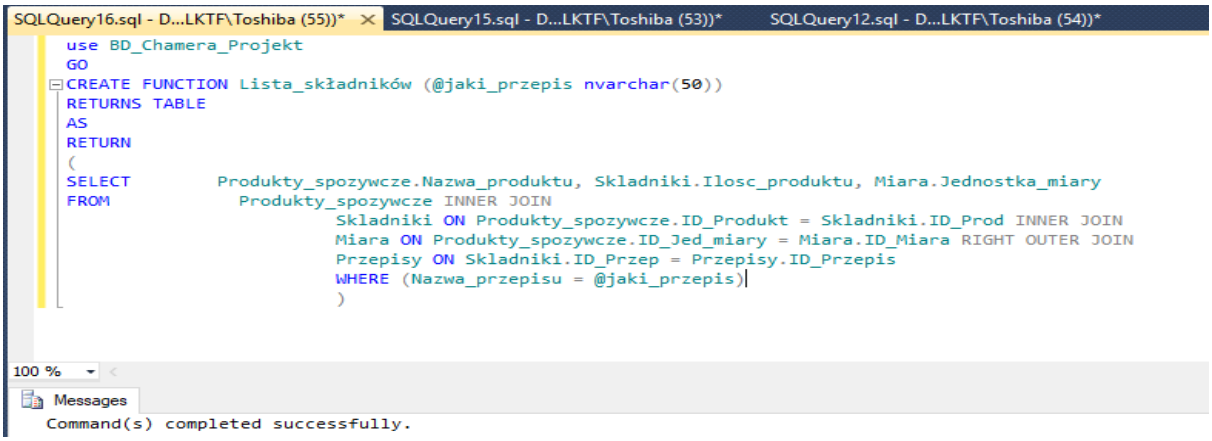
dbo.Przepisy.ID_Kuchnia = dbo.Rodzaj_kuchni.ID_Rodzaj_kuchni

dbo.Przepisy.ID_Poz_trudnosci = dbo.Poziom_trudnosci.ID_ Poziom

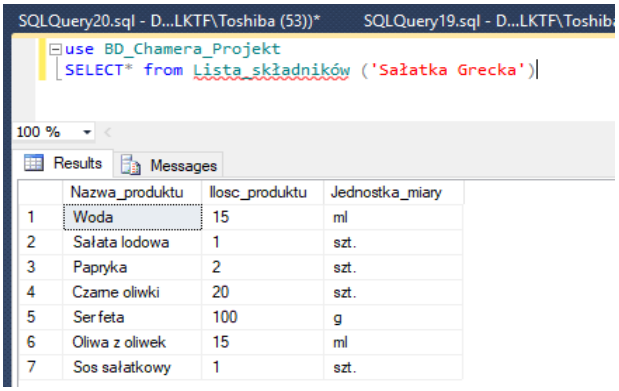
Zadania bazy:

1. **FUNKCJA - „Lista_składników'**

   Po wpisaniu interesującego nas przepisu wyszukane zostają wszystkie składniki potrzebne do jego realizacji wraz z podaniem ilości oraz jednostki w jakiej są wyrażone produkty.



Realizacja:



	Nazwa_produktu	Ilosc_produktu	Jednostka_miary
1	Woda	15	ml
2	Sałata lodowa	1	szt.
3	Papryka	2	szt.
4	Czarne oliwki	20	szt.
5	Ser feta	100	g
6	Oliwa z oliwek	15	ml
7	Sos sałatkowy	1	szt.

Funkcja zwróciła w formie tabeli listę składników dla przepisu o nazwie „Sałatka Grecka".

2. FUNKCJA – „przepis_z”

Służy do wyszukiwania przepisów zawierających dany produkt spożywczy na swojej liście składników.

```
use BD_Chamera_Projekt
go
CREATE FUNCTION przepis_z(@szukany_produkt nvarchar(50))
RETURNS TABLE
AS
RETURN
(
SELECT        Przepisy.Nazwa_przepisu
FROM            Produkty_spozywcze INNER JOIN
                         Skladniki ON Produkty_spozywcze.ID_Produkt = Skladniki.ID_Prod INNER JOIN
                         Przepisy ON Skladniki.ID_Przep = Przepisy.ID_Przepis
WHERE (Nazwa_produktu = @szukany_produkt)
)
```

```
Command(s) completed successfully.
```

Realizacja:

```
use BD_Chamera_Projekt
SELECT* from przepis_z ('Cukier')
```

	Nazwa_przepisu
1	Magdalenki
2	Trifle

Wyszukanie przepisów zawierających jako składnik cukier.

3. FUNKCJA - „danie_z_kuchni”

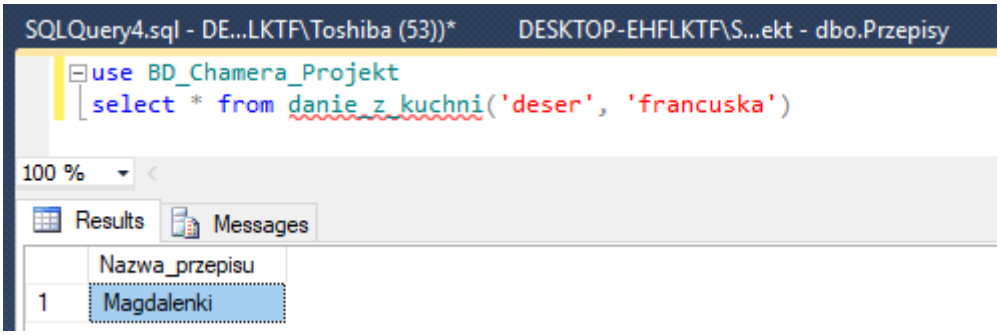
Służy do wyszukiwania określonego typu dania ze wskazaniem preferencji co do pochodzenia potrawy – rodzaju kuchni.

```
use BD_Chamera_Projekt
go
CREATE FUNCTION danie_z_kuchni(@jakie_danie nvarchar(20), @jaka_kuchnia nvarchar(25))
RETURNS TABLE
AS
RETURN
(
SELECT        Przepisy.Nazwa_przepisu
FROM            Przepisy RIGHT OUTER JOIN
                         Typ_dania ON Przepisy.ID_Danie = Typ_dania.ID_Typ_dania RIGHT OUTER JOIN
                         Rodzaj_kuchni ON Przepisy.ID_Kuchnia = Rodzaj_kuchni.ID_Rodzaj_kuchni

WHERE (Nazwa_typu_dania  = @jakie_danie)and (Nazwa_Kuchni = @jaka_kuchnia)
)
```

```
Command(s) completed successfully.
```

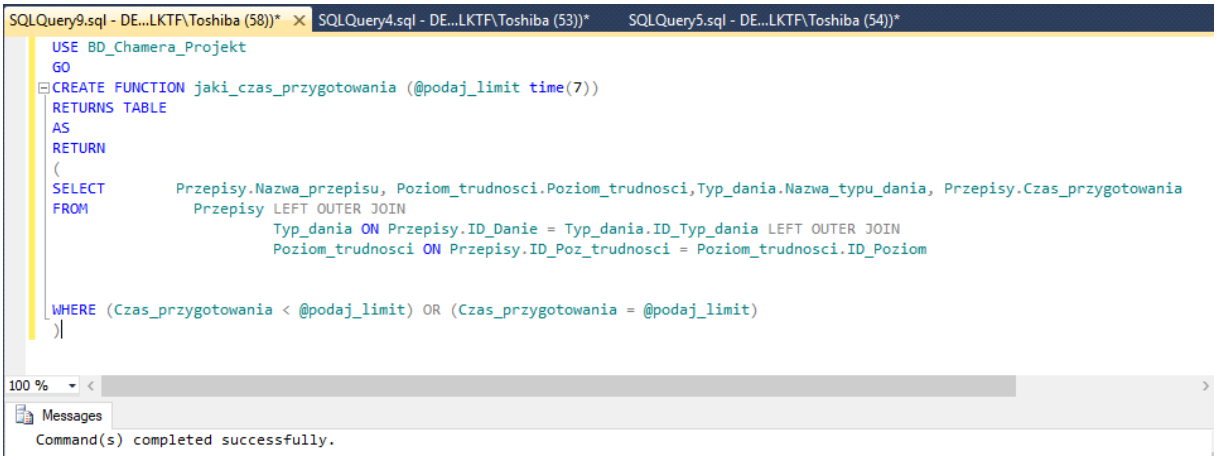
Realizacja:



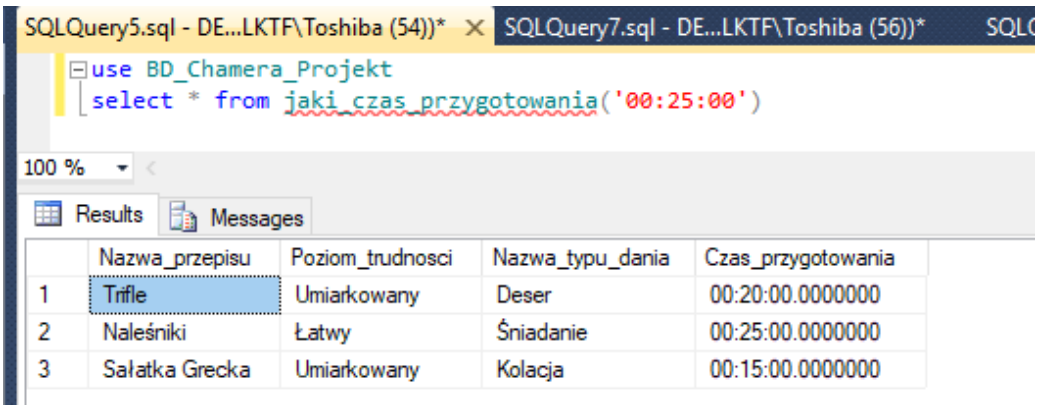
Wyszukanie deserów kuchni francuskiej.

4. **FUNKCJA – „jaki_czas_przygotowania”**

   Służy do wyszukiwania przepisów, których czas przygotowania jest mniejszy bądź równy temu wskazanemu przez nas.



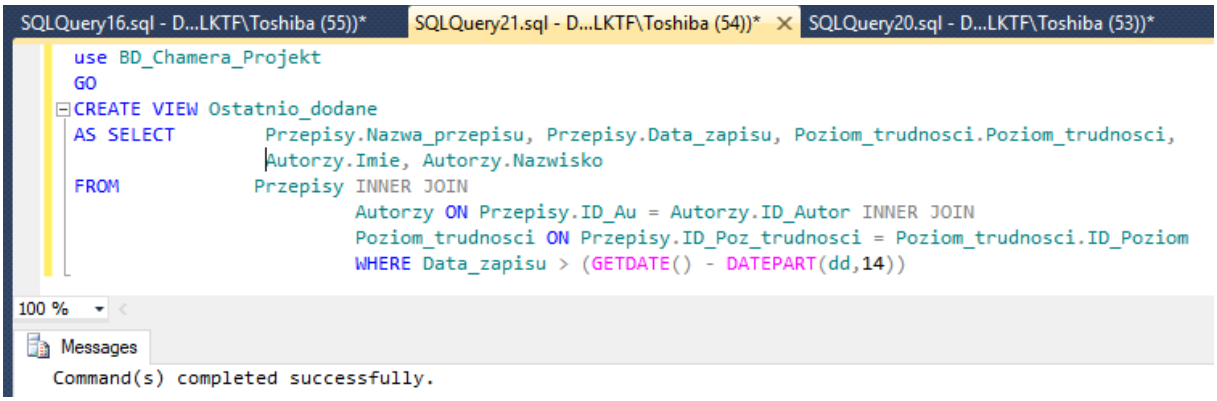
Realizacja:



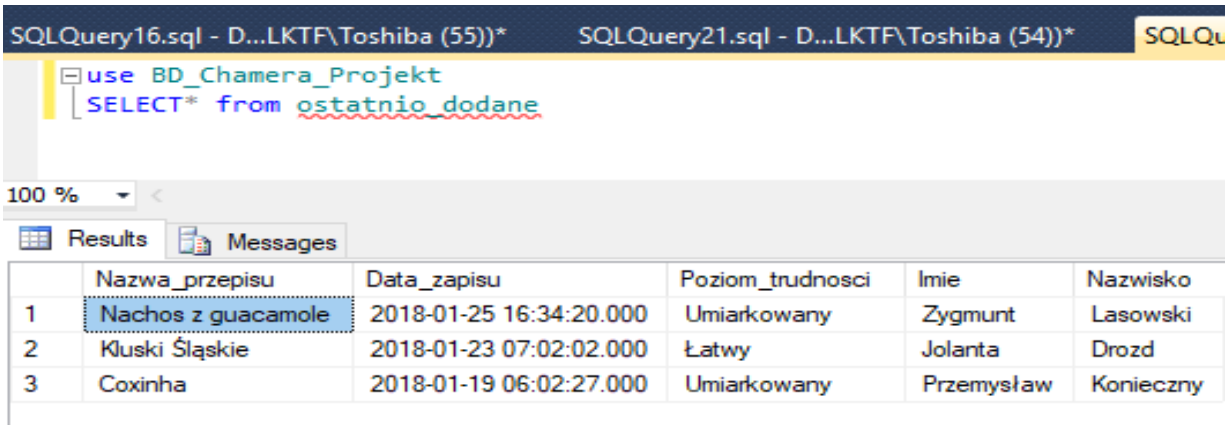
Wyszukanie przepisów, których czas przygotowania nie przekracza 25 minut.

5. WIDOK – „Ostatnio_dodane”

Wyświetla przepisy, które zostały dodane do bazy w przeciągu ostatnich dwóch tygodni.



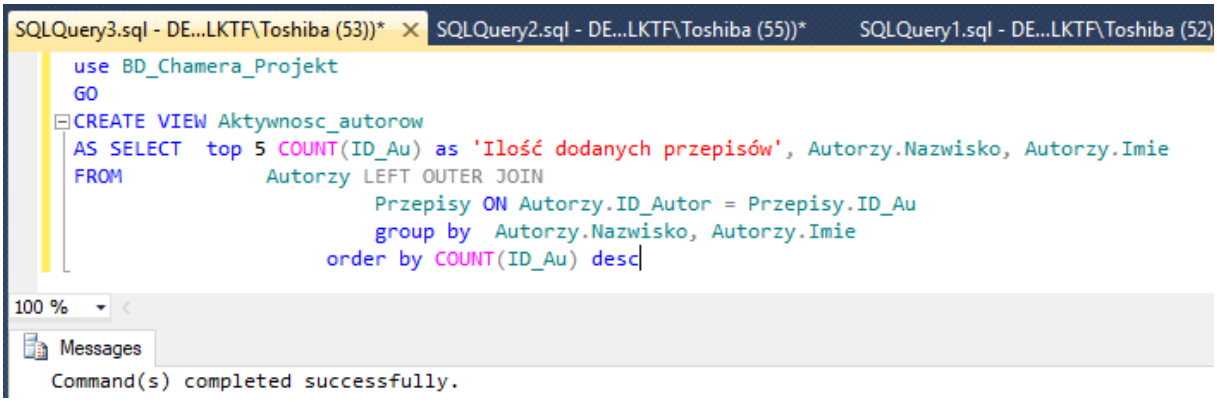
Realizacja:



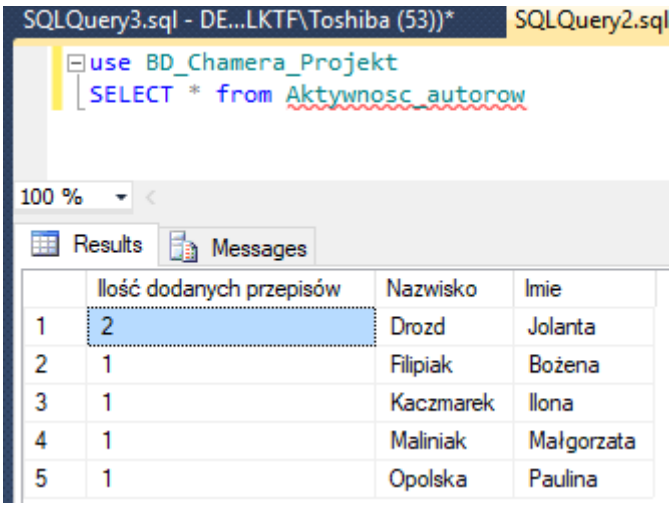
	Nazwa_przepisu	Data_zapisu	Poziom_trudnosci	Imie	Nazwisko
1	Nachos z guacamole	2018-01-25 16:34:20.000	Umiarkowany	Zygmunt	Lasowski
2	Kluski Śląskie	2018-01-23 07:02:02.000	Łatwy	Jolanta	Drozd
3	Coxinha	2018-01-19 06:02:27.000	Umiarkowany	Przemysław	Konieczny

6. WIDOK – „Aktywnosc_autorow”

Wyświetla ranking pięciu autorów, którzy dodali najwięcej przepisów do bazy.



Realizacja:

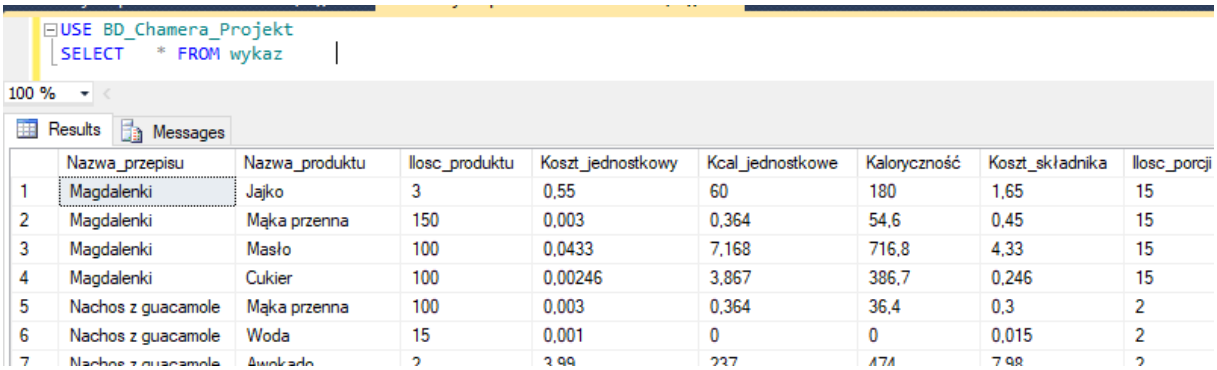


	Ilość dodanych przepisów	Nazwisko	Imie
1	2	Drozd	Jolanta
2	1	Filipiak	Bożena
3	1	Kaczmarek	Ilona
4	1	Maliniak	Małgorzata
5	1	Opolska	Paulina

7. WIDOK – „wykaz” (pomocniczy – użyty przy funkcji nr 8, 9 i 10)

Wyświetla informacje dotyczące skłądników w danym przepisie – ich kaloryczność i koszty jednostkowe jak i w przeliczeniu na potrzebną do przepisu ilość.

Realizacja:



	Nazwa_przepisu	Nazwa_produktu	Ilosc_produktu	Koszt_jednostkowy	Kcal_jednostkowe	Kaloryczność	Koszt_składnika	Ilosc_porcji
1	Magdalenki	Jajko	3	0,55	60	180	1,65	15
2	Magdalenki	Mąka przenna	150	0,003	0,364	54,6	0,45	15
3	Magdalenki	Masło	100	0,0433	7,168	716,8	4,33	15
4	Magdalenki	Cukier	100	0,00246	3,867	386,7	0,246	15
5	Nachos z guacamole	Mąka przenna	100	0,003	0,364	36,4	0,3	2
6	Nachos z guacamole	Woda	15	0,001	0	0	0,015	2
7	Nachos z guacamole	Awokado	2	3,99	237	474	7,98	2

8. FUNKCJA – „Koszty_porcji”

Funkcja służy do wyszukania kosztu jednej porcji podanego przepisu.

```
use BD_Chamera_Projekt
GO
CREATE FUNCTION Koszty_porcji (@Szukany_przepis nvarchar(60))
RETURNS FLOAT
AS
BEGIN
DECLARE @x FLOAT , @y float
SELECT @x=sum(Koszt_składnika)from wykaz where Nazwa_przepisu=@Szukany_przepis
SELECT @y=(Ilosc_porcji) from wykaz where Nazwa_przepisu=@Szukany_przepis
RETURN round(@x/@y, 2)
END
```

```
Command(s) completed successfully.
```

Realizacja:

```
use BD_Chamera_Projekt
SELECT dbo.Koszty_porcji ('Pizza') as koszt
```

	koszt
1	8,75

Sprawdzono ile wynosi koszt porcji pizzy.

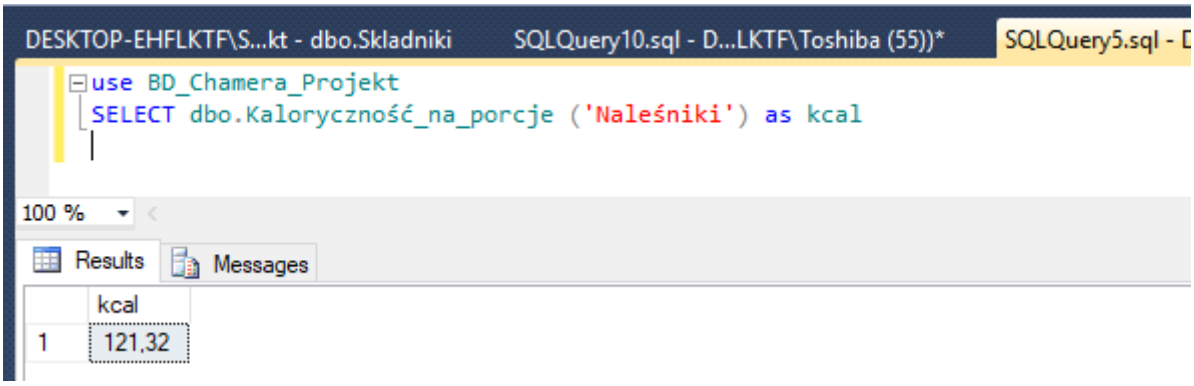
9. FUNKCJA – „Kaloryczność_na_porcje”

Umożliwia sprawdzenie kaloryczności potrawy w przeliczeniu na porcje po podaniu nazwy sprawdzanego przepisu.

```
use BD_Chamera_Projekt
GO
CREATE FUNCTION Kaloryczność_na_porcje (@Sprawdzany_przepis nvarchar(60))
RETURNS FLOAT
AS
BEGIN
DECLARE @a FLOAT , @b float
SELECT @a=sum(Kaloryczność)from wykaz where Nazwa_przepisu=@Sprawdzany_przepis
SELECT @b=(Ilosc_porcji) from wykaz where Nazwa_przepisu=@Sprawdzany_przepis
RETURN @a/@b
END
```

```
Command(s) completed successfully.
```

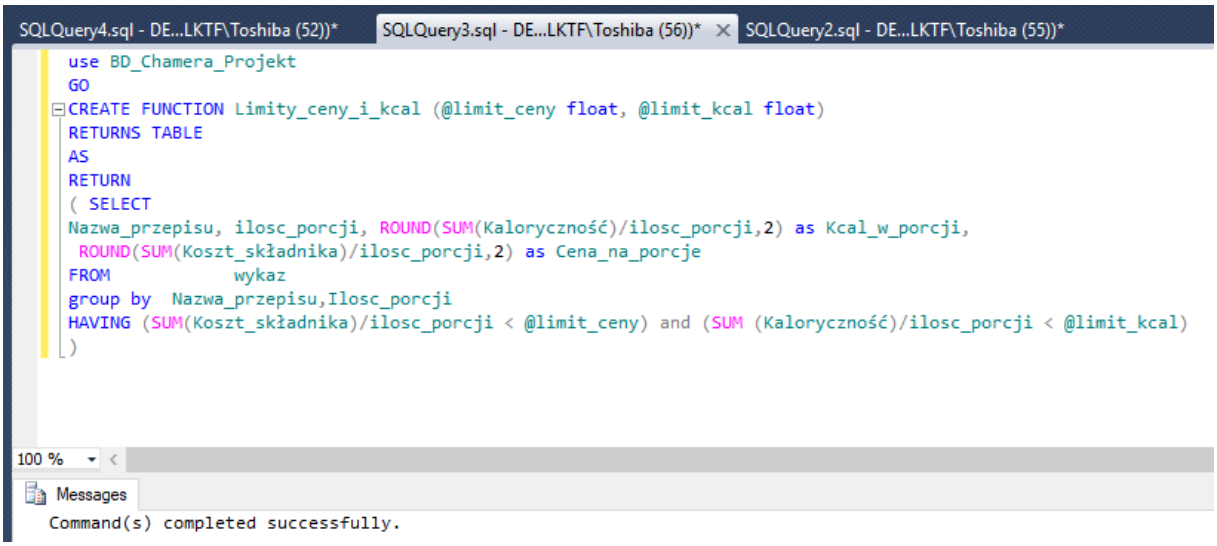
Realizacja:



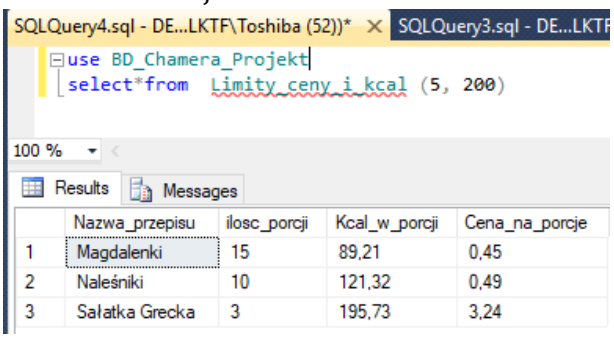
Sprawdzono ile kalorii posiada porcja naleśników.

10. FUNKCJA – „Limity_ceny_i_kcal”

Służy do wyszukiwania tych przepisów, których cena i kaloryczność w przeliczeniu na porcje są niższe od wskazanych w zapytaniu limitów.



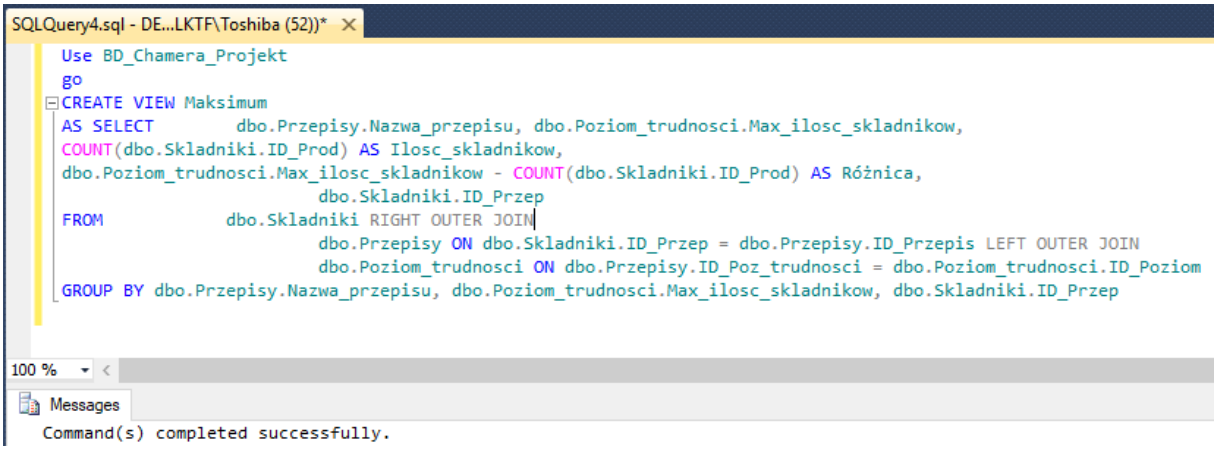
Realizacja:



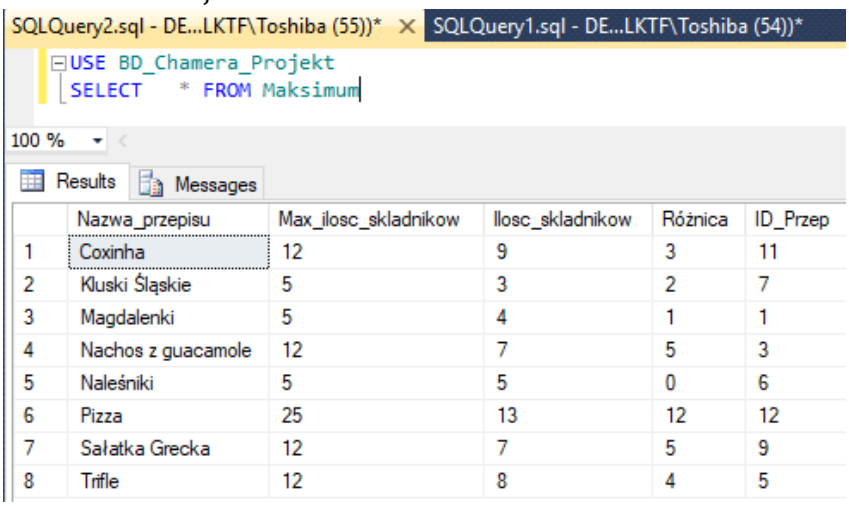
Wyszukano przepisów o cenie mniejszej niż 5 zł i kaloryczności poniżej 200kcal na porcję.

11. WIDOK – „Maksimum” (pomocniczy – użyty w triggerze nr 12)

Zawiera informacje dotyczące ilości wprowadzonych składników do danego przepisu, a także ile można jeszcze wprowadzić (poz. Różnica) mając na uwadze maksymalną ilość składników dla danego poziomu trudności.



Realizacja:



	Nazwa_przepisu	Max_ilosc_skladnikow	Ilosc_skladnikow	Różnica	ID_Przep
1	Coxinha	12	9	3	11
2	Kluski Śląskie	5	3	2	7
3	Magdalenki	5	4	1	1
4	Nachos z guacamole	12	7	5	3
5	Naleśniki	5	5	0	6
6	Pizza	25	13	12	12
7	Sałatka Grecka	12	7	5	9
8	Trifle	12	8	4	5

12. TRIGGER - „dodawanie_skladnikow”

Sprawdzanie czy kolejny wpisywany składnik nie przekracza limitu dozwolonego na dany poziom trudności. W przypadku jego przekroczenia blokuje możliwość zapisania danych i zwraca komunikat o tym informujący.

SQLQuery5.sql - DE...LKTF\Toshiba (53))*

```
USE BD_Chamera_Projekt
GO
CREATE TRIGGER [dbo].[dodawanie_skladnikow] ON skladniki

AFTER INSERT, UPDATE AS
DECLARE @c int, @d int
SELECT @c=inserted.ID_Przep FROM INSERTED
SELECT @d=Różnica from Maksimum where ID_Przep=@c
IF @d > 0 or @d = 0 PRINT 'Składnik wprowadzono poprawnie. Pozostała liczba składników do wprowadzenia = ' + cast(@d as varchar)
ELSE
BEGIN
PRINT 'Składnik nie może zostać dodany! Przekroczono w tym przepisie limit produktów na dany poziom trudności.'
ROLLBACK
END
```

100 %

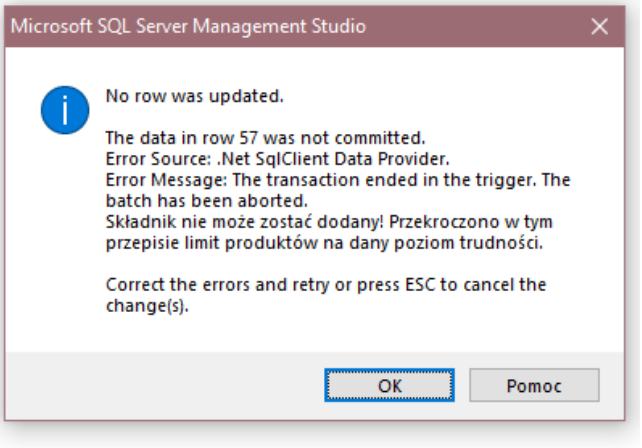
Messages

```
Command(s) completed successfully.
```

Realizacja:

DESKTOP-EHFLKTF\S...kt - dbo.Skladniki | SQLQuery12.sql - D...LKTF\Toshiba (54))*

	ID_Przep	ID_Prod	Ilosc_produktu
	12	2	250
	12	3	150
	12	6	5
	12	8	1
	12	9	10
	12	11	125
	12	1018	1
	12	1019	25
	12	1028	1
	12	1029	50
	12	1030	75
	12	1031	6
	12	1032	75
	6	16	20
*	NULL	NULL	NULL



Po próbie dodania nowego składnika do listy przepisu „Naleśniki” (ID_Przep=6) zablokowano wprowadzanie rekordu i wyświetlono komunikat.