1. Изучите, что такое [ограничения](https://webformyself.com/sql-ogranicheniya/#:~:text=%D0%9E%D1%82%20%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D0%B0%3A%20%D0%BE%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20SQL%20%E2%80%94%20%D1%8D%D1%82%D0%BE,%D1%81%D1%82%D0%BE%D0%BB%D0%B1%D1%86%D0%BE%D0%B2%2C%20%D0%BB%D0%B8%D0%B1%D0%BE%20%D0). Какие ограничения есть для таблицы Person.Address?

|  |  |  |
| --- | --- | --- |
| **CONSTRAINT** | **COLUMN** | **DESCRIPTION** |
| **NOT NULL** | AddressID |  |
| AddressLine1 |  |
| City |  |
| StateProvinceID |  |
| PostalCode |  |
| rowguid |  |
| ModifiedDate |  |
| **PRIMARY KEY** | AddressID | Primary key for Address records |
| **FOREIGN KEY** | StateProvinceID | Unique identification number for the state or province. Foreign key to StateProvince table |
| **DEFAULT** |  | |
| **[DF\_Address\_ModifiedDate]** | ModifiedDate | Default constraint value of GETDATE() |
| **[DF\_Address\_rowguid]** | rowguid] | Default constraint value of NEWID() |

1. В базе данных AdventureWorks2017 создать таблицу Patients для ведения наблюдений за температурой пациентов больницы. Таблица должна содержать поля:

**ID** – числовое поле. Авто заполняется.

**FirstName** – имя пациента, может быть пустым.

**LastName** – фамилия пациента, не может быть пустым.

**SSN** – уникальный идентификатор пациента.

**Email** – электронная почта пациента. Формируется по следующему правилу: первая большая буква FirstName + маленькие 3 буквы LastName + @mail.com (например, Akli@mail.com). Полезная ссылка [здесь](https://docs.microsoft.com/en-us/sql/relational-databases/tables/specify-computed-columns-in-a-table?view=sql-server-ver15#:~:text=A%20computed%20column%20is%20a,in%20SQL%20Server%202019%20(15.).

**Temp** – температура пациента, значения не должны превышать 45.

**CreatedDate** — дата измерений.

CREATE TABLE [AdventureWorks2017].[dbo].[Patients](

[ID] int IDENTITY(1,1),

[FirstName] nvarchar(20) NULL,

[LastName] nvarchar(100) NOT NULL,

[SSN] uniqueidentifier ROWGUIDCOL CONSTRAINT [DF\_Patients\_SSN] DEFAULT (newid()),

[Email] nvarchar(13),

[Temp] float CHECK (([Temp]>=(34.5) AND [Temp]<=(45))),

[CreatedDate] datetime NOT NULL

)

ALTER TABLE [dbo].[Patients] DROP COLUMN [Email]

ALTER TABLE [dbo].[Patients] ADD [Email] AS (CONCAT(LEFT(UPPER([FirstName]),1), LEFT(LOWER([LastName]),3), '@mail.com')) PERSISTED

GO

*ВОПРОСЫ:*

1. *Когда лучше (в каких случаях) использовать float, decimal, numeric. Например, поле температура – какой лучше тип данных?*
2. *Вопрос по полю email - при таком правиле могут быть совпадающие email, а этого наверное быть не должно? Как предусмотреть? Если такой адрес есть, то добавить например ещё одну букву? А если поставить на поле UNIQUE, то невозможно будет вставить ещё одну запись с таком email?*
3. Добавить в таблицу несколько произвольных записей. Убедиться, что ограничения действуют. Посмотрите на ошибки, если вставка противоречит ограничениям.

INSERT INTO [dbo].[Patients]

(

[FirstName],

[LastName],

[Temp],

[CreatedDate]

)

VALUES

(

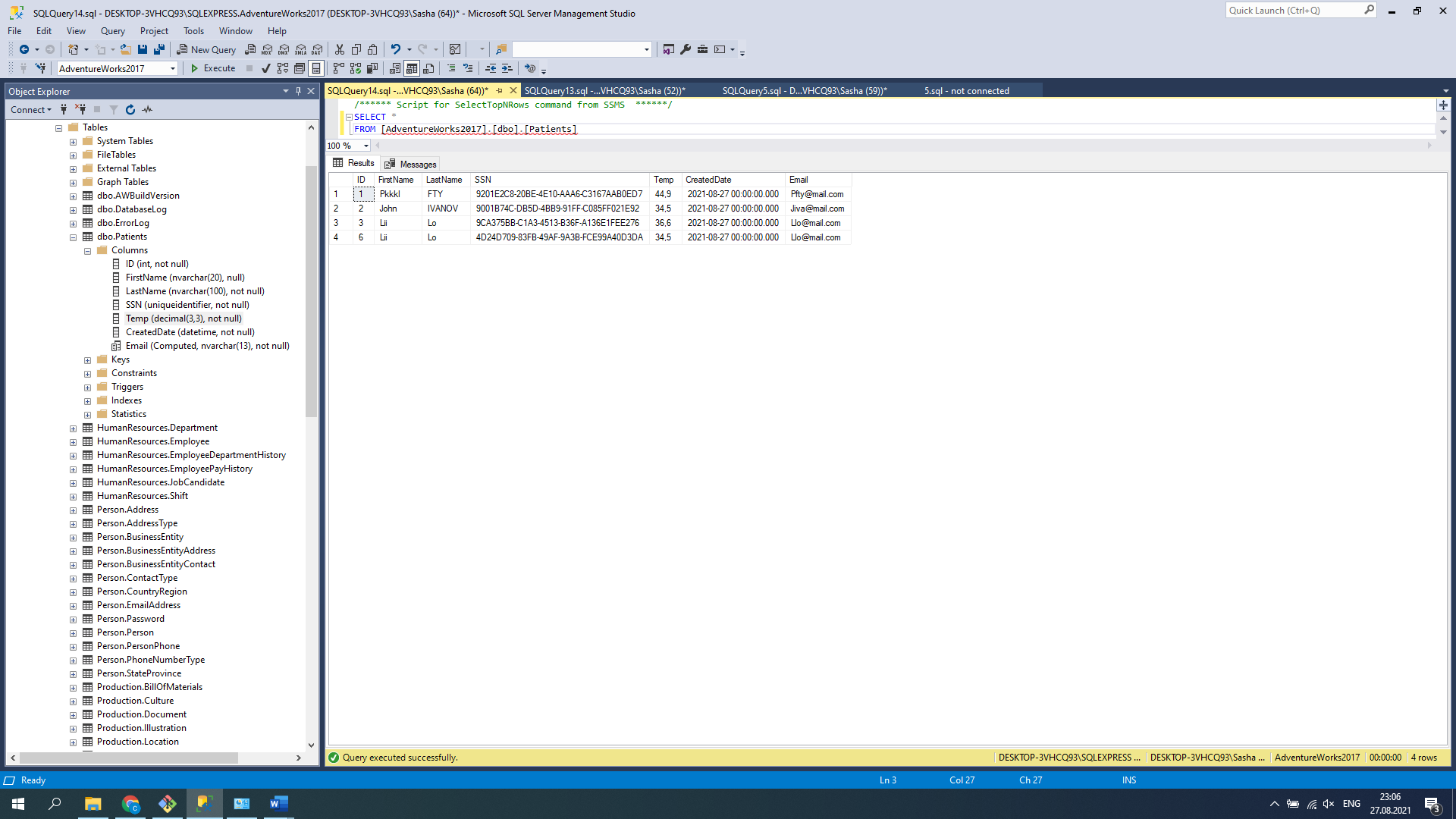
N 'Lii',

N 'Lo',

34.5,

'27.08.2021'

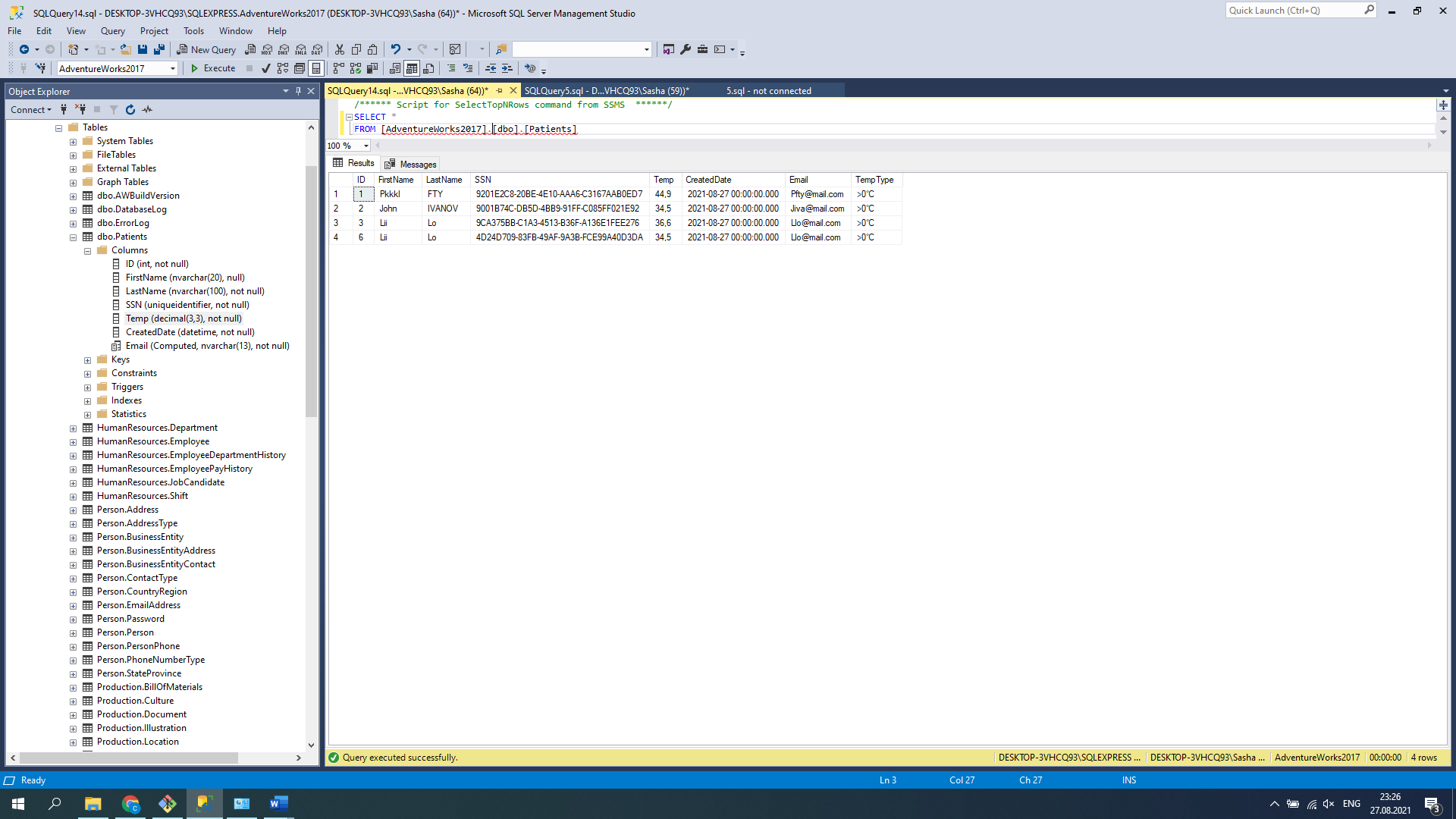
)



1. Добавить поле TempType со следующими значениями ‘< 0°C’, ‘> 0°C’ на основе значений из поля Temp ( используйте ALTER TABLE ADD column AS ). Посмотрите на данные, которые получились.

ALTER TABLE [dbo].[Patients] ADD [TempType] AS (CASE WHEN [Temp] > 0 THEN '>0°C' ELSE '<0°C' END) PERSISTED

GO



1. Создать представление Patients\_v, показывающее температуру в градусах Фаренгейта (°F = °Cx9/5 + 32)

CREATE VIEW Patients\_v

AS

SELECT [ID]

,[FirstName]

,[LastName]

,[SSN]

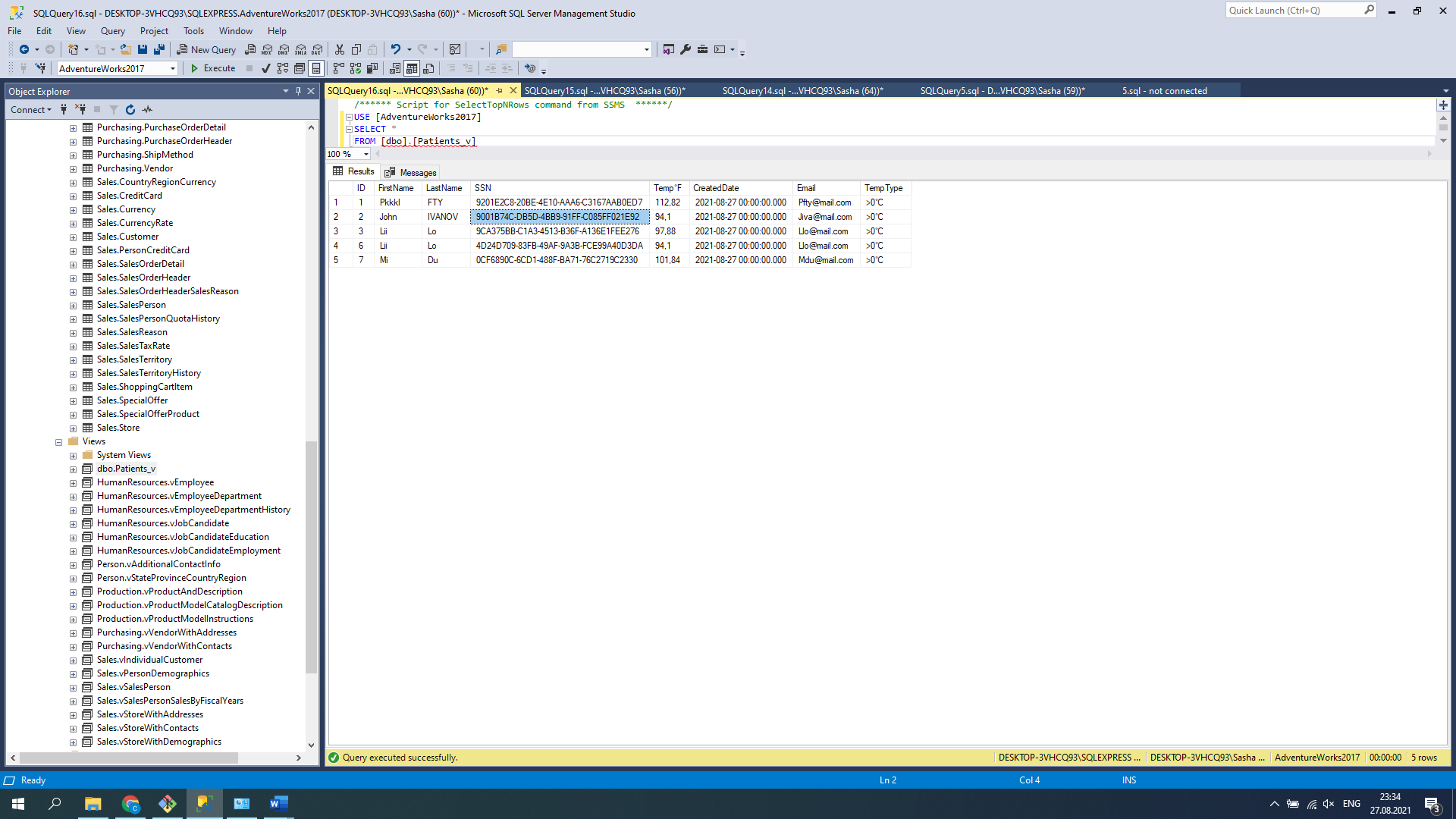
,[Temp°F] = [Temp]\*9/5 + 32

,[CreatedDate]

,[Email]

,[TempType]

FROM [AdventureWorks2017].[dbo].[Patients]



1. Создать функцию, которая возвращает температуру в градусах Фаренгейта, при подаче на вход градусы в Цельсиях.

CREATE FUNCTION [dbo].fn\_DegreeFahrenheit (@DegreeCelcius float)

RETURNS float

AS

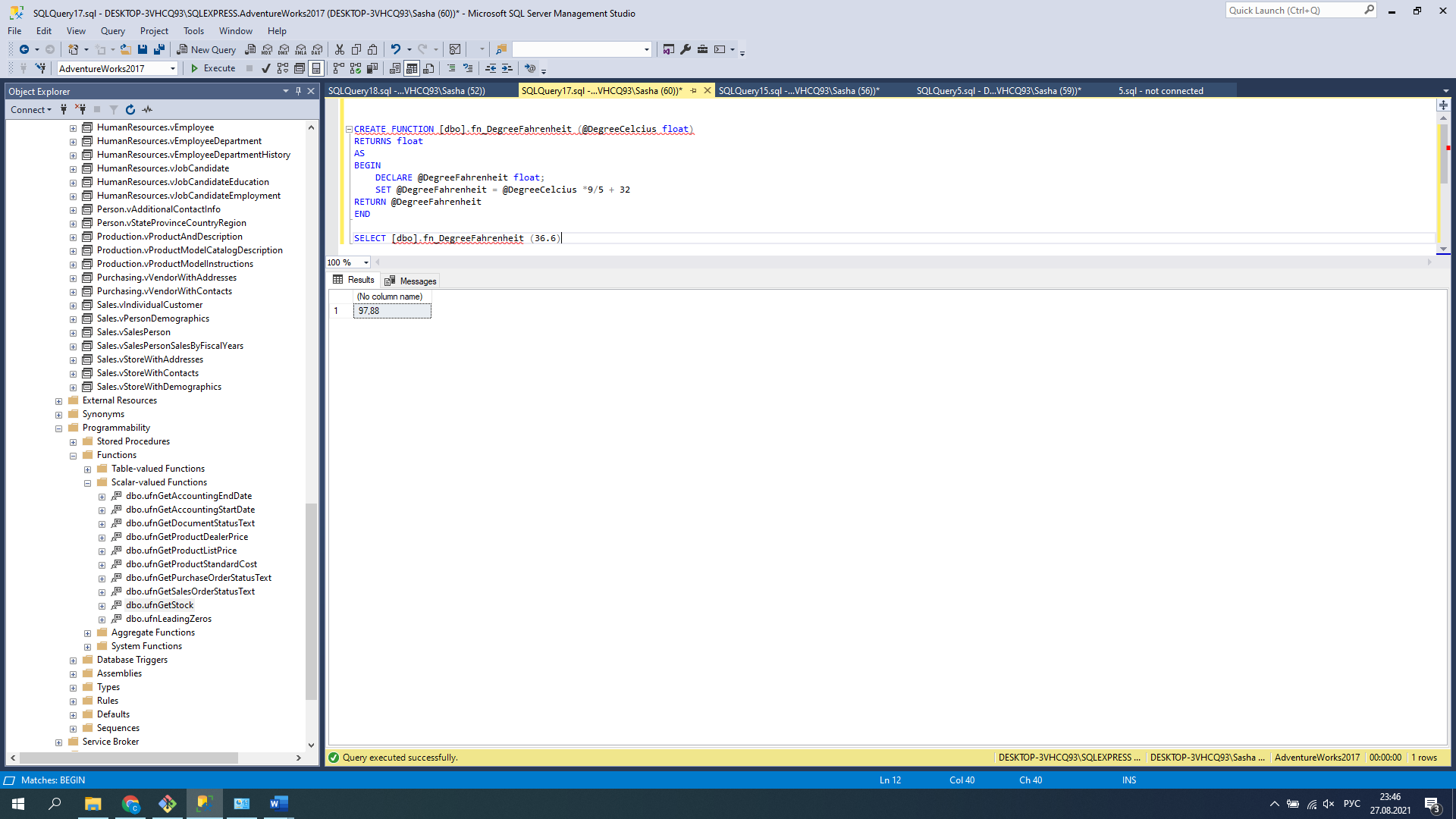
BEGIN

DECLARE @DegreeFahrenheit float;

SET @DegreeFahrenheit = @DegreeCelcius \*9/5 + 32

RETURN @DegreeFahrenheit

END



1. Перепишите решение задачи g из прошлого дз с использованием переменной, максимально упрощая select.

DECLARE @Date date = '22.08.2021'

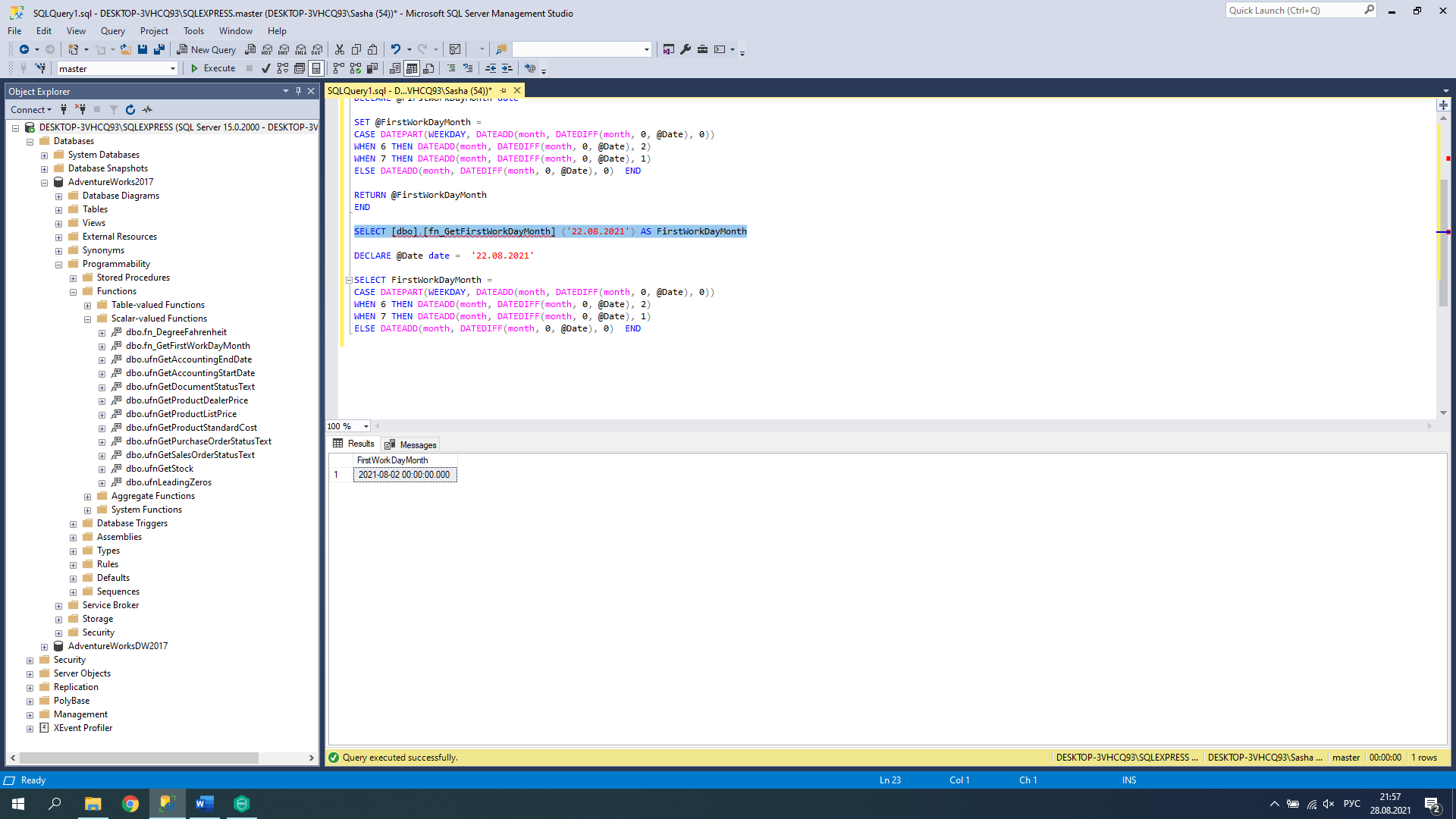
SELECT FirstWorkDayMonth =

CASE DATEPART(WEEKDAY, DATEADD(month, DATEDIFF(month, 0, @Date), 0))

WHEN 6 THEN DATEADD(month, DATEDIFF(month, 0, @Date), 2)

WHEN 7 THEN DATEADD(month, DATEDIFF(month, 0, @Date), 1)

ELSE DATEADD(month, DATEDIFF(month, 0, @Date), 0) END



**Или создать скалярную функцию**

CREATE FUNCTION [dbo].[fn\_GetFirstWorkDayMonth] (@Date date)

RETURNS date

AS

BEGIN

DECLARE @FirstWorkDayMonth date

SET @FirstWorkDayMonth =

CASE DATEPART(WEEKDAY, DATEADD(month, DATEDIFF(month, 0, @Date), 0))

WHEN 6 THEN DATEADD(month, DATEDIFF(month, 0, @Date), 2)

WHEN 7 THEN DATEADD(month, DATEDIFF(month, 0, @Date), 1)

ELSE DATEADD(month, DATEDIFF(month, 0, @Date), 0) END

RETURN @FirstWorkDayMonth

END

**Проверяем**

SELECT [dbo].[fn\_GetFirstWorkDayMonth] ('22.08.2021') AS FirstWorkDayMonth

