**Домашняя работа 5**

SQL Database

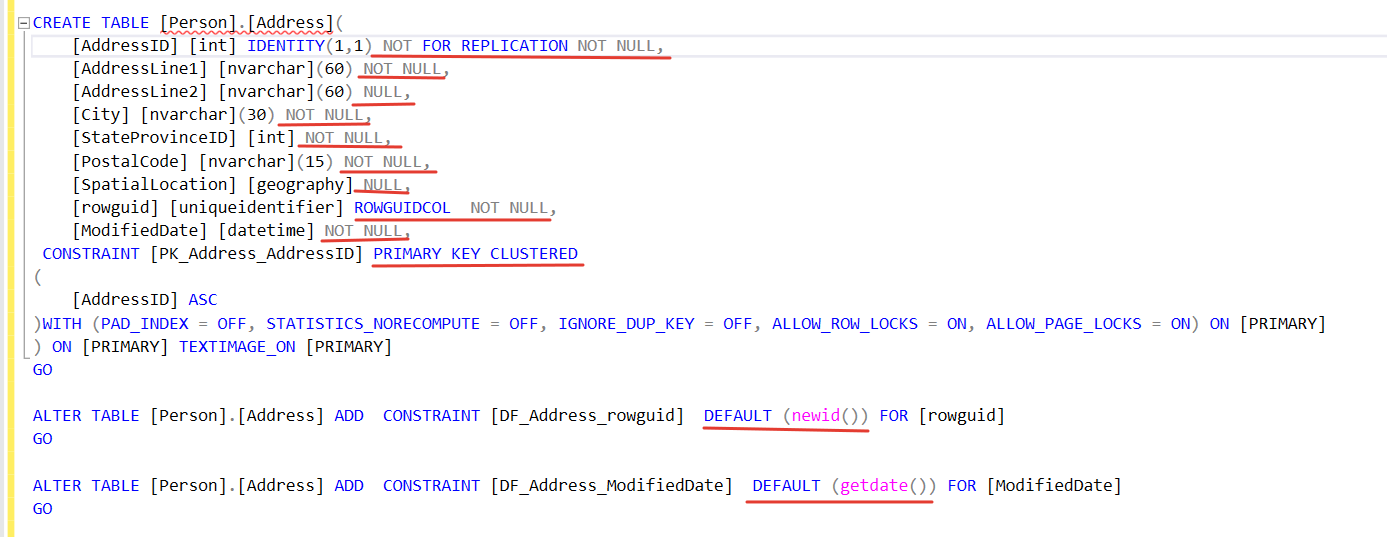
**Задания**

1. Изучите новый тип данных - [uniqueidentifier](https://docs.microsoft.com/ru-ru/sql/t-sql/data-types/uniqueidentifier-transact-sql?view=sql-server-ver15). Запустите следующий код несколько раз, посмотрите результаты.

DECLARE @myid uniqueidentifier = NEWID();

SELECT CONVERT(CHAR(255), @myid) AS 'char';

1. Познакомьтесь с новой функцией [Newid()](https://docs.microsoft.com/en-us/sql/t-sql/functions/newid-transact-sql?view=sql-server-ver15). Разберите примеры после статьи. Запустите скрипты на своей базе данных. Посмотрите, что получилось.
2. Посмотрите как функции getdate() и newid() используются для формирования таблицы Production.Document.
3. Изучите [свойство identity](https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-table-transact-sql-identity-property?view=sql-server-ver15). Рассмотрите пример A. Посмотрите, как используется данное свойство при формировании таблицы Production.Product.
4. Изучите объект базы данных - [Триггер](https://professorweb.ru/my/sql-server/2012/level3/3_18.php). Сделайте [практическую часть](https://metanit.com/sql/sqlserver/12.2.php), повторяя за автором.
5. 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?



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 (используйте функцию).

**Email** – электронная почта пациента. Формируется по следующему правилу: большая первая 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.

USE [AdventureWorks2017]

CREATE TABLE Patients

(

ID int IDENTITY(1,1),

FirstName nvarchar(30) NULL,

LastName nvarchar(100) NOT NULL,

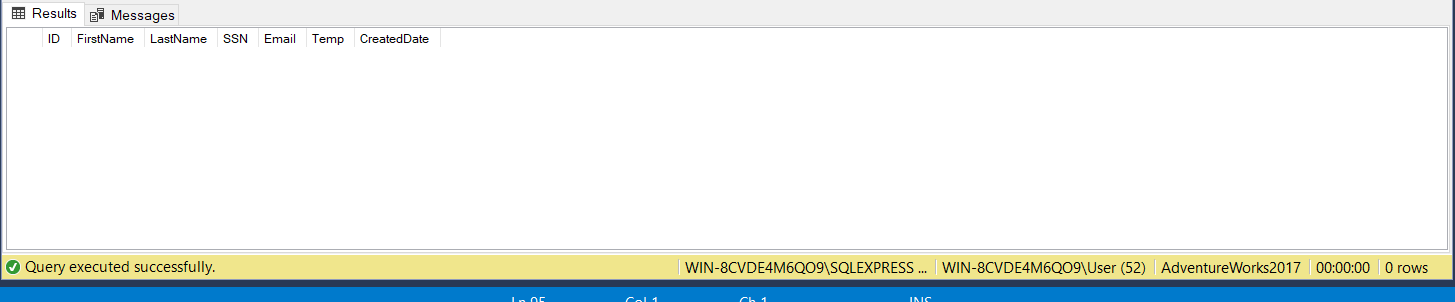
SSN uniqueidentifier NOT NULL DEFAULT (newid()),

Email AS (CONCAT (UPPER (LEFT (FirstName, 1)), LOWER (LEFT(LastName, 3)),'@gmail.com')),

Temp decimal (3,1) CHECK (([Temp]>=(34.5) AND [Temp]<=(45))),

CreatedDate datetime NOT NULL

)



1. Добавить в таблицу несколько произвольных записей. Убедиться, что ограничения действуют. Посмотрите на ошибки, если вставка противоречит ограничениям.

INSERT INTO [dbo].[Patients]

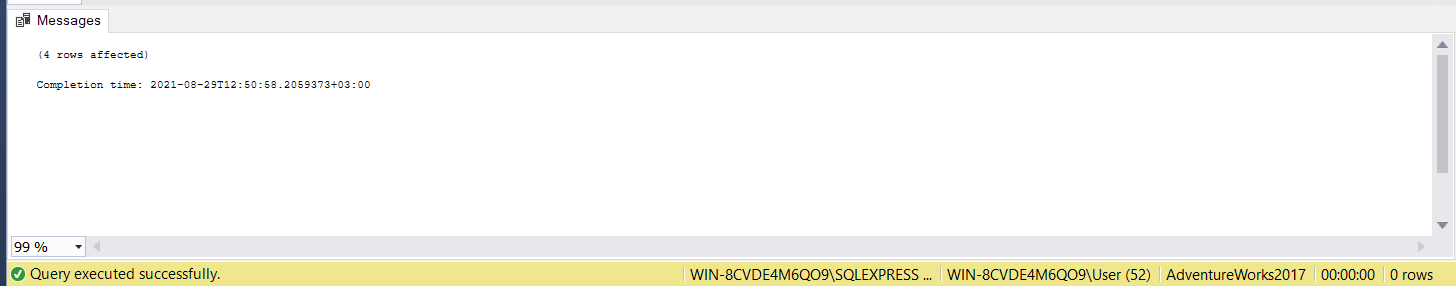
( [FirstName], [LastName], [Temp], [CreatedDate])

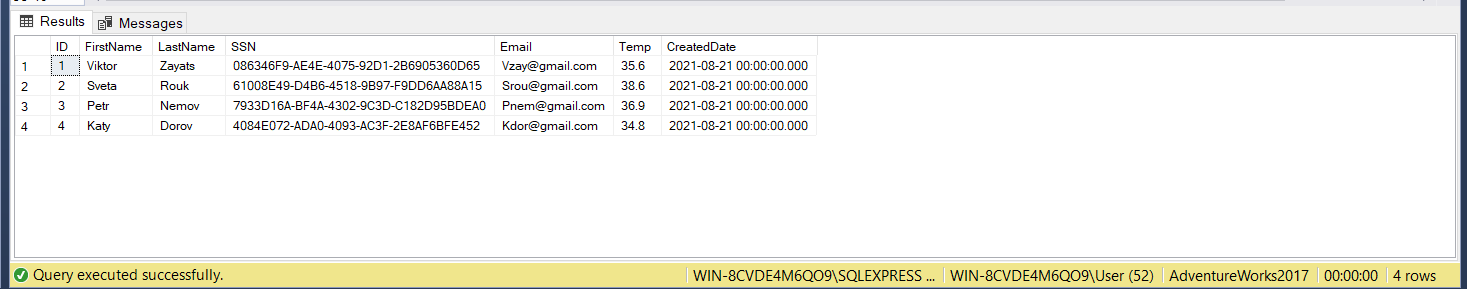
VALUES ( 'Viktor','Zayats', 35.6, '2021-08-21'),

( 'Sveta','Rouk', 38.6, '2021-08-21'),

( 'Petr','Nemov', 36.9, '2021-08-21'),

( 'Katy','Dorov', 34.8, '2021-08-21')





1. Добавить поле TempType со следующими значениями ‘< 38°C’, ‘> =38°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 (

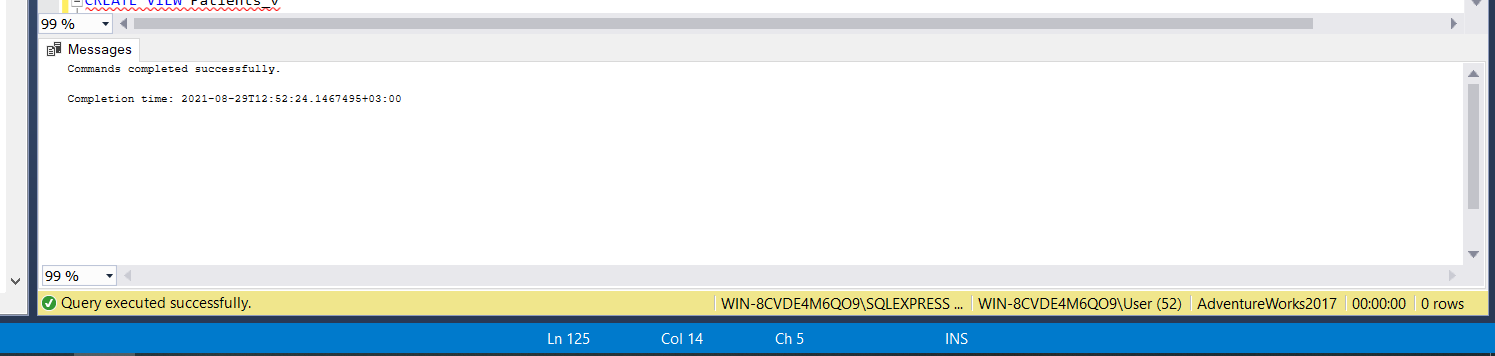
CASE

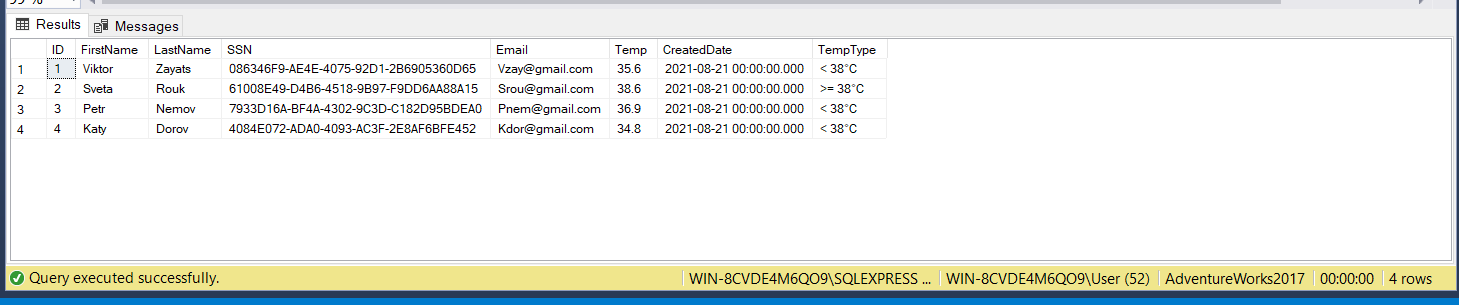
WHEN [Temp] < 38 THEN '< 38°C'

WHEN [Temp] >= 38 THEN '>= 38°C'

END

)





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] \* 9 / 5 + 32 AS [Temp°F],

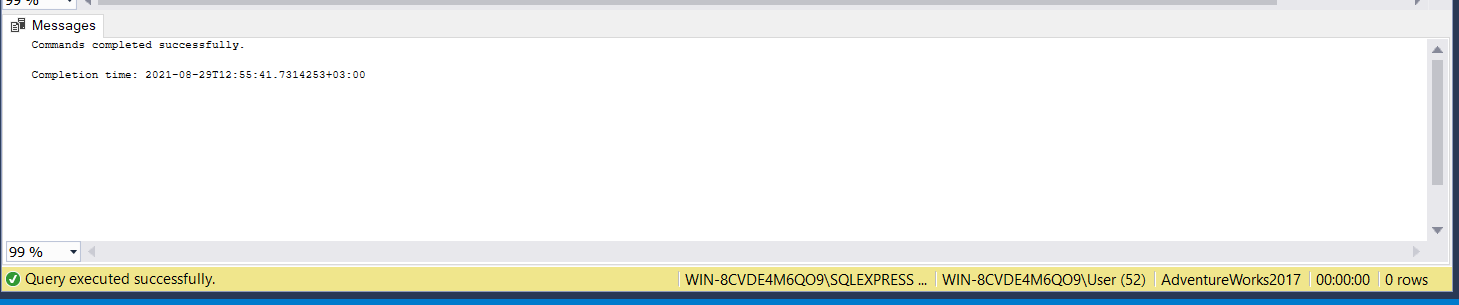
[CreatedDate],

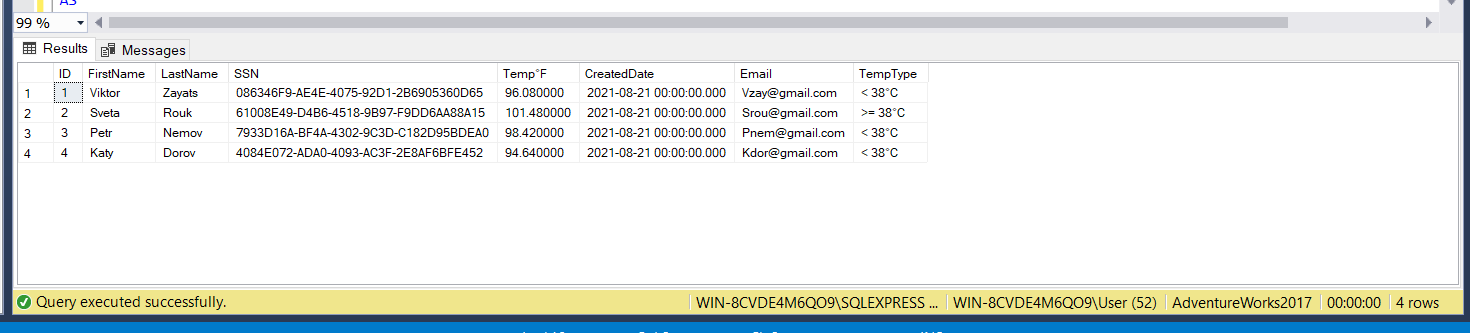
[Email],

[TempType]

FROM

[AdventureWorks2017].[dbo].[Patients]





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

CREATE FUNCTION dbo.TempF (

@TempF DECIMAL(3,1)

)

RETURNS DECIMAL(12,2)

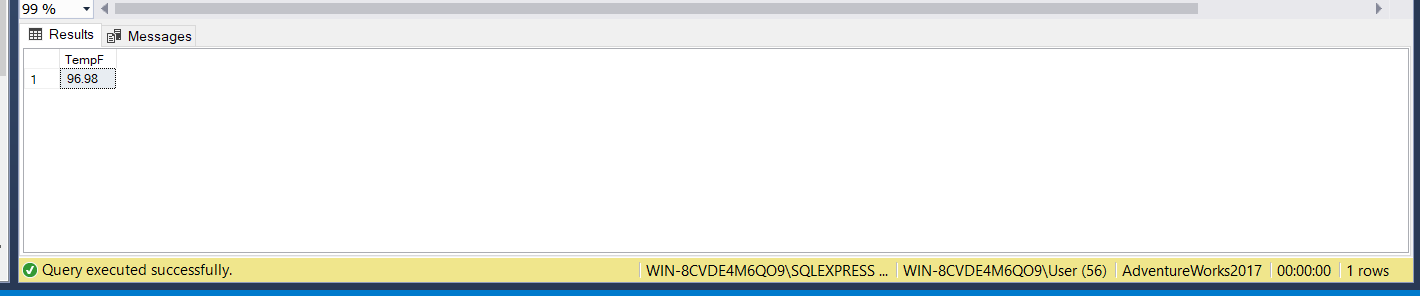
AS

BEGIN

RETURN @TempF \* 9 / 5 + 32

END;

SELECT dbo.TempF (36.1) AS TempF



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

DECLARE @Date INT

SET @Date = DATEPART(dd, GETDATE())

SELECT

CASE

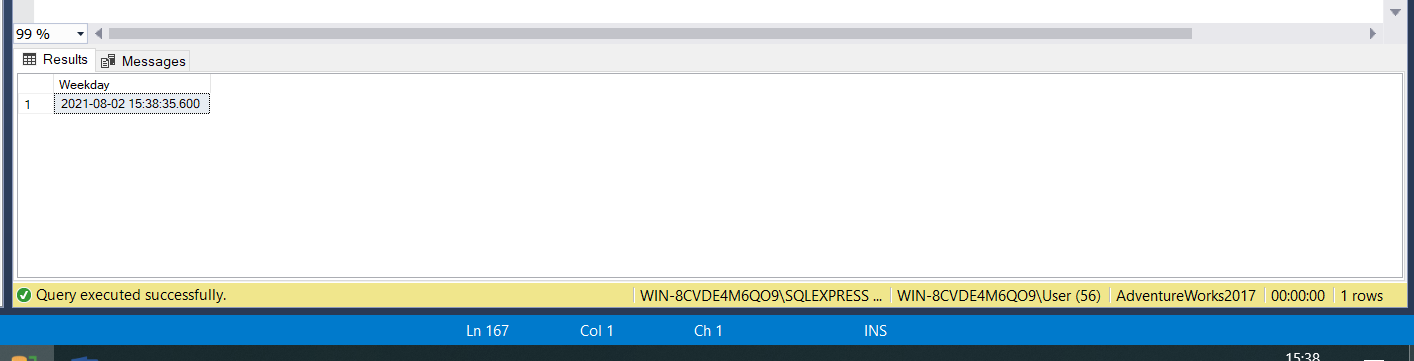
WHEN DATENAME(dw, DATEADD(dd, - @Date + 1, GETDATE())) = 'Saturday'

THEN DATEADD(dd, - @Date + 3, GETDATE())

WHEN DATENAME(dw, DATEADD(dd, - @Date + 1, GETDATE())) = 'Sunday'

THEN DATEADD(dd, - @Date + 2, GETDATE())

ELSE DATEADD(dd, - @Date + 1, GETDATE())

END 'Weekday'