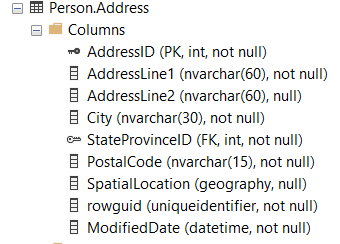
6. Изучите, что такое [ограничения](https://webformyself.com/sql-ogranicheniya/#:~:text=%D0%9E%D1%82%20%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D0%B0%3A%20%D0%BE%D0%B3%D1%80%D0%B0%D0%BD%D0%B8%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20SQL%20%E2%80%94%20%D1%8D%D1%82%D0%BE,%D1%81%D1%82%D0%BE%D0%BB%D0%B1%D1%86%D0%BE%D0%B2%2C%20%D0%BB%D0%B8%D0%B1%D0%BE%20%D0).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

Ограничения SQL — это правила, применяемые к столбцам данных таблицы. Они используются, чтобы ограничить типы данных, которые могут храниться в таблице. Это обеспечивает точность и надежность данных в базе данных



* **NOT NULL** - гарантирует, что столбец не может иметь значение NULL
* **PRIMARY KEY** - комбинация NOT NULL и UNIQUE. Уникально идентифицирует каждую строку в таблице
* **FOREIGN KEY** - однозначно идентифицирует строку / запись в другой таблице
* **CHECK** - обеспечивает, чтобы все значения в столбце удовлетворяли конкретному условию
* **DEFAULT** - устанавливает значение по умолчанию для столбца, если не указано значение
* **INDEX** - используется для быстрого создания и извлечения данных из базы данных

7. 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.

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

CREATE TABLE Patients

(

ID INT IDENTITY (1,1) NOT NULL,

FirstName NVARCHAR(30) NOT NULL,

LastName NVARCHAR(30) NOT NULL,

SSN UNIQUEIDENTIFIER NOT NULL DEFAULT NEWID(),

Email NVARCHAR(200),

Temp INT, CHECK (Temp<45),

CreatedData DATE

);

SELECT \*

FROM Patients

INSERT INTO Patients (FirstName, LastName,CreatedData, Temp)

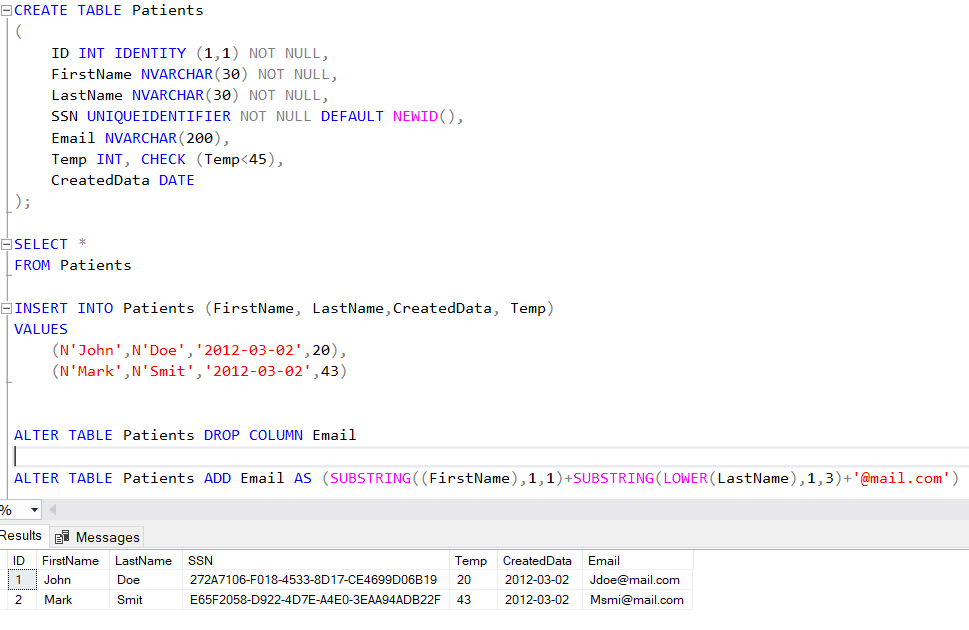
VALUES

(N'John',N'Doe','2012-03-02',20),

(N'Mark',N'Smit','2012-03-02',43)

ALTER TABLE Patients DROP COLUMN Email

ALTER TABLE Patients ADD Email AS (SUBSTRING((FirstName),1,1)+SUBSTRING(LOWER(LastName),1,3)+'@mail.com')



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.

CREATE TABLE Patients

(

ID INT IDENTITY (1,1) NOT NULL,

FirstName NVARCHAR(30) NOT NULL,

LastName NVARCHAR(30) NOT NULL,

SSN UNIQUEIDENTIFIER NOT NULL DEFAULT NEWID(),

Email NVARCHAR(200),

Temp INT, CHECK (Temp<45),

CreatedData DATE

);

SELECT \*

FROM Patients

INSERT INTO Patients (FirstName, LastName,CreatedData, Temp)

VALUES

(N'John',N'Doe','2012-03-02',20),

(N'Mark',N'Smit','2012-03-02',43)

ALTER TABLE Patients DROP COLUMN Email

ALTER TABLE Patients ADD Email AS (SUBSTRING((FirstName),1,1)+SUBSTRING(LOWER(LastName),1,3)+'@mail.com')

INSERT INTO Patients (FirstName, LastName,CreatedData, Temp)

VALUES

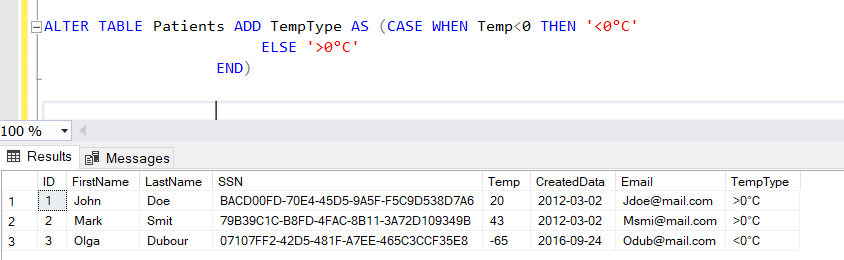
(N'Olga',N'Dubour','2016-09-24',-65)

ALTER TABLE Patients ADD TempType AS (CASE WHEN Temp<0 THEN '<0°C'

ELSE '>0°C'

END)

Результат:

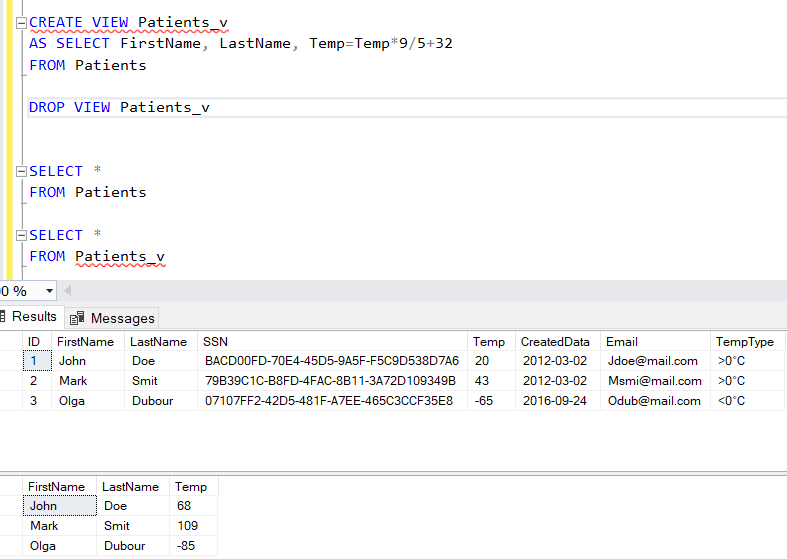


1. Создать представление Patients\_v, показывающее температуру в градусах Фаренгейта (°F = °Cx9/5 + 32)

CREATE VIEW Patients\_v

AS SELECT FirstName, LastName, Temp=Temp\*9/5+32

FROM Patients



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

CREATE FUNCTION Fahrenheit(@tempp int)

RETURNS int

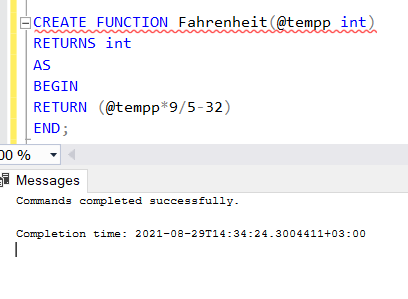
AS

BEGIN

RETURN (@tempp\*9/5-32)

END;

SELECT Fahrenheit(20)



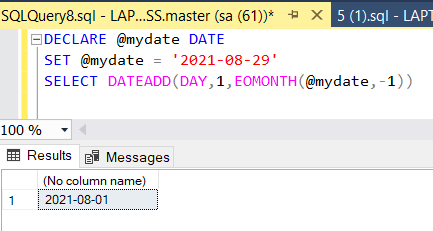
1. Перепишите решение задачи g из прошлого дз с использованием переменной, максимально упрощая select.

g.Найдите первый будний день месяца (FROM не используем). Нужен стандартный код на все времена.

DECLARE @mydate DATE

SET @mydate = '2021-08-29'

SELECT DATEADD(DAY,1,EOMONTH(@mydate,-1))



DECLARE @mydate DATE

SET @mydate = '2021-08-29'

SELECT Dateadd(dd, -Datepart(dd, @mydate) + 2, @mydate)

SELECT Datename(dw, Dateadd(dd, -Datepart(dd, Getdate()) + 2, Getdate()))

