1. **При каких значениях оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат?**

Одинаковый результат будет, если данные уникальны, т.е при группировке в каждой группе будет 1 запись.

1. Решите на базе данных AdventureWorks2017 следующие задачи.
2. **Изучите данные в таблице Production.UnitMeasure. Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. Сколько всего различных кодов здесь есть?**

/\*Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. Сколько всего различных кодов здесь есть?\*/

USE AdventureWorks2017

-- есть несколько вариантов, не знаю какой оптимальный.

--вариант 1 проверяю есть ли вообще хоть одна запись [UnitMeasureCode], которая начинается на Т, если есть, вывожу True, иначе False

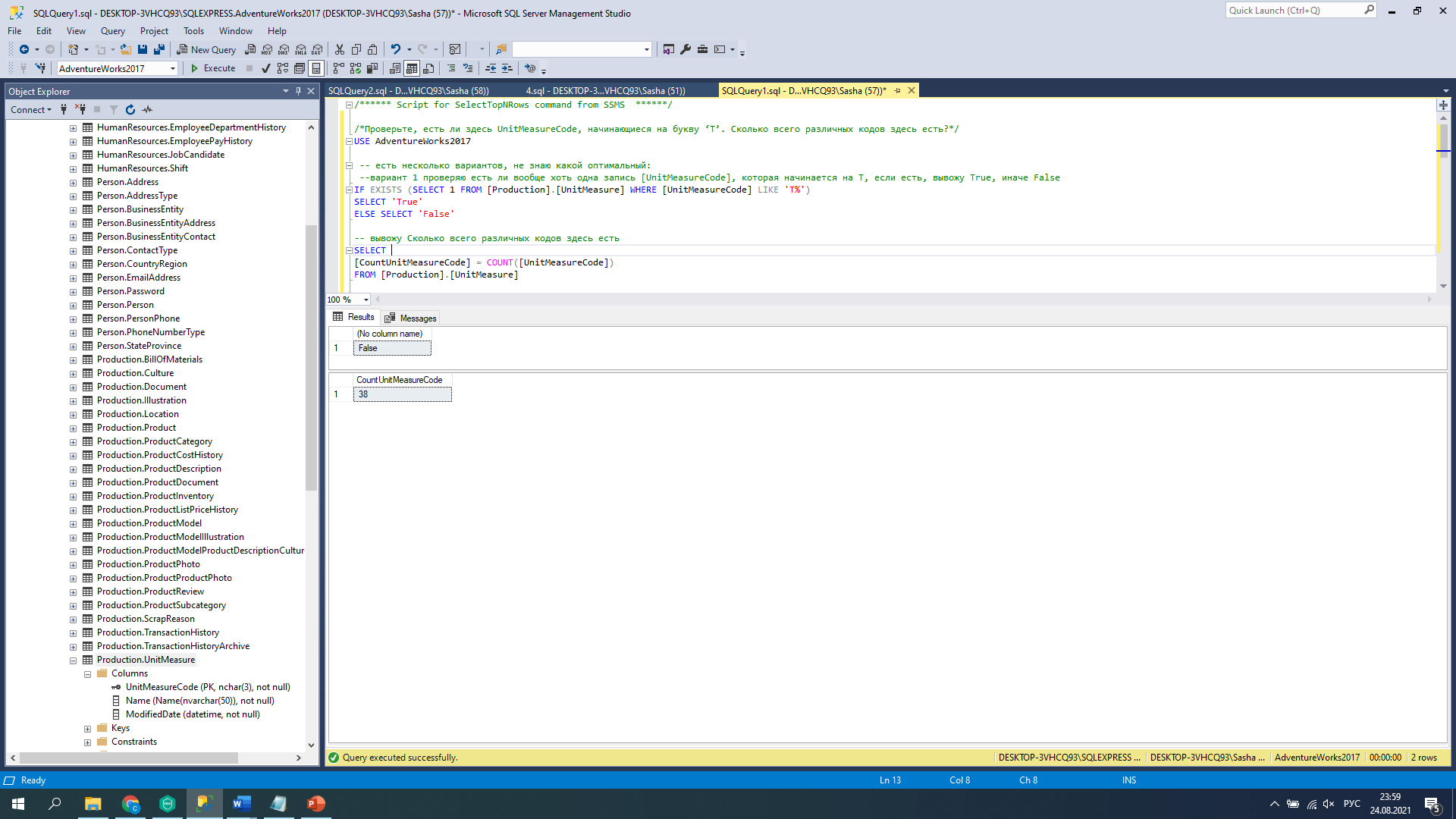
SELECT IIF((EXISTS (SELECT 1 FROM [Production].[UnitMeasure] WHERE [UnitMeasureCode] LIKE 'T%' )), 'True' ,'False') AS Exist

-- вывожу Сколько всего различных кодов здесь есть

SELECT

[CountUnitMeasureCode] = COUNT([UnitMeasureCode])

FROM [Production].[UnitMeasure]



--вариант 2 в одном запросе через SELECT

SELECT

[CountUnitMeasureCode\_Begin\_T] = (SELECT COUNT([UnitMeasureCode])

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'Т%'),

[CountUnitMeasureCode] = COUNT([UnitMeasureCode])

FROM [Production].[UnitMeasure]

-- вариант 3 в одном запросе через CASE

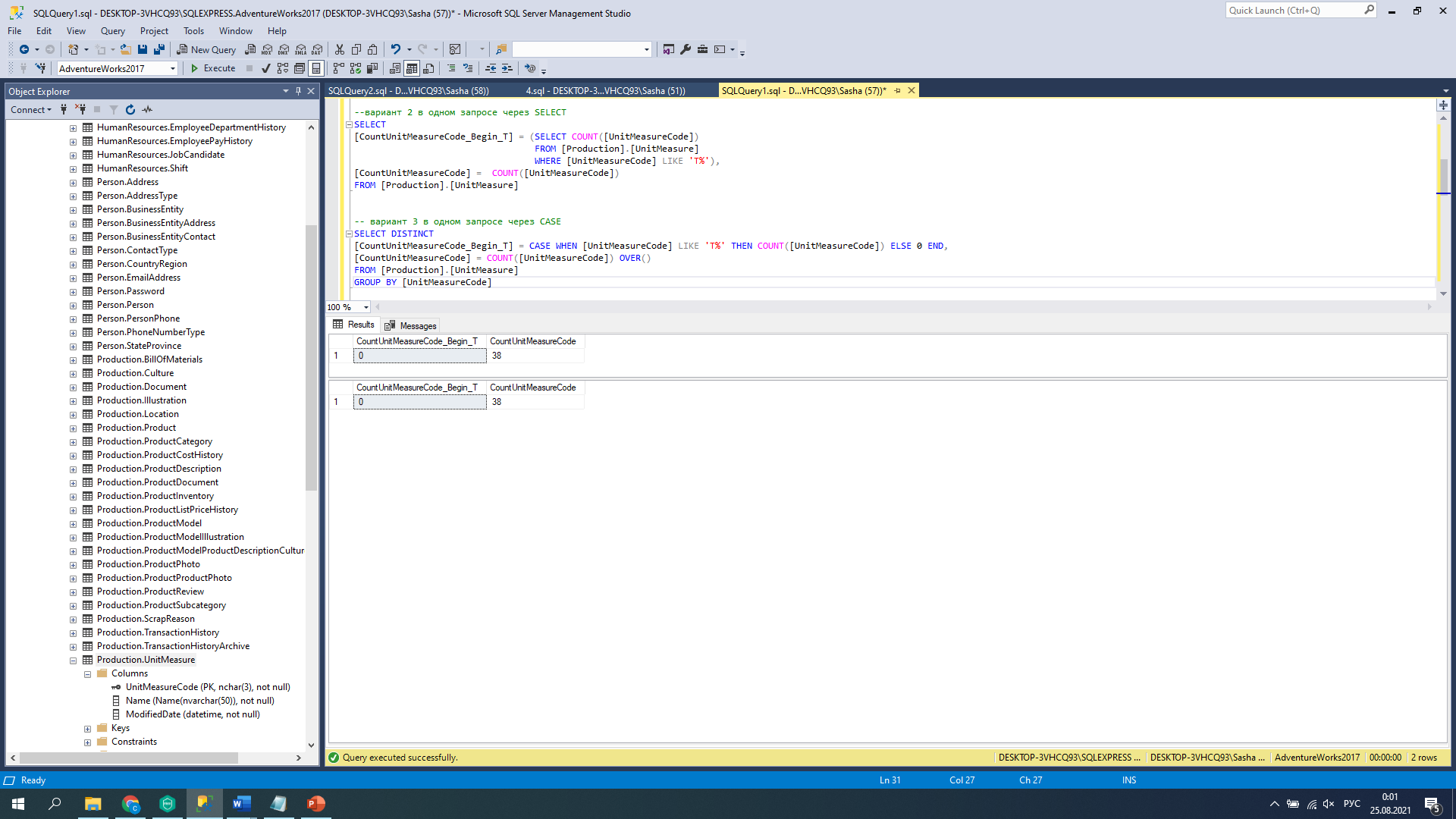
SELECT DISTINCT

[CountUnitMeasureCode\_Begin\_T] = CASE WHEN [UnitMeasureCode] LIKE 'Т%' THEN COUNT([UnitMeasureCode]) ELSE 0 END,

[CountUnitMeasureCode] = COUNT([UnitMeasureCode]) OVER()

FROM [Production].[UnitMeasure]

GROUP BY [UnitMeasureCode]



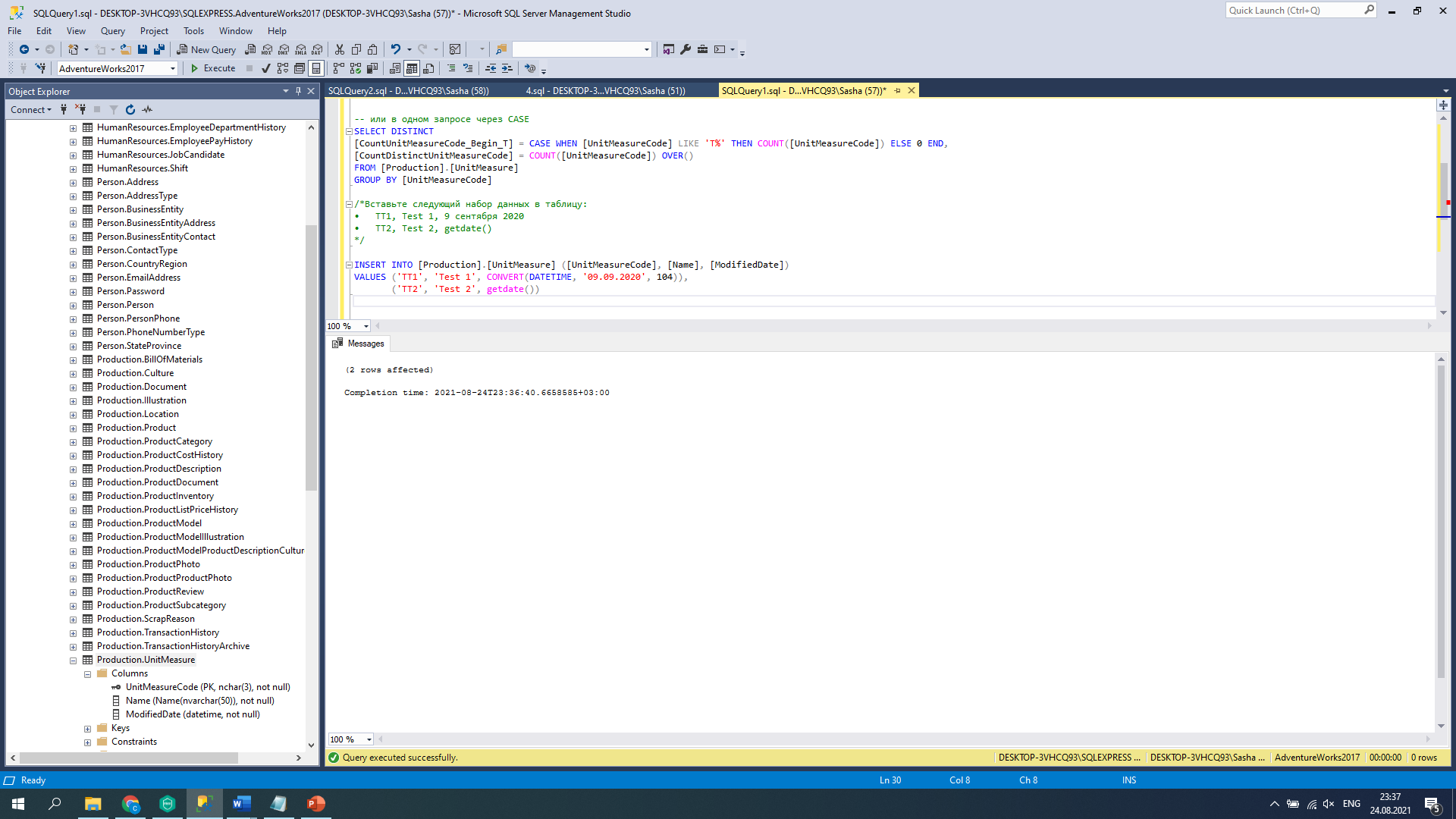
Вставьте следующий набор данных в таблицу:

* TT1, Test 1, 9 сентября 2020
* TT2, Test 2, getdate()

INSERT INTO [Production].[UnitMeasure] ([UnitMeasureCode], [Name], [ModifiedDate])

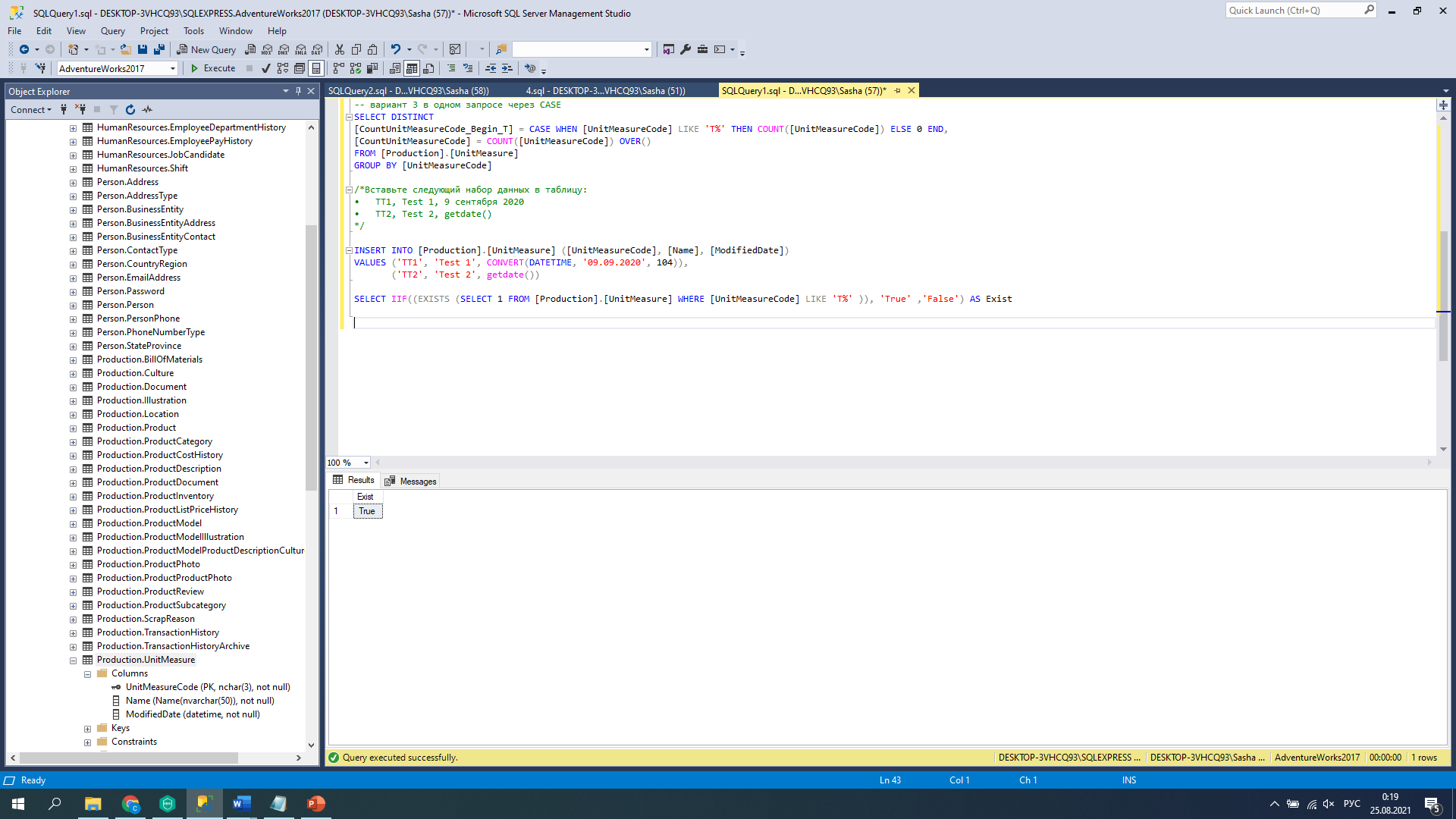
VALUES ('TT1', 'Test 1', CONVERT(DATETIME, '09.09.2020', 104)),

('TT2', 'Test 2', getdate())



Проверьте теперь, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’.

SELECT IIF((EXISTS (SELECT 1 FROM [Production].[UnitMeasure] WHERE [UnitMeasureCode] LIKE 'T%' )), 'True' ,'False') AS Exist



1. Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest.

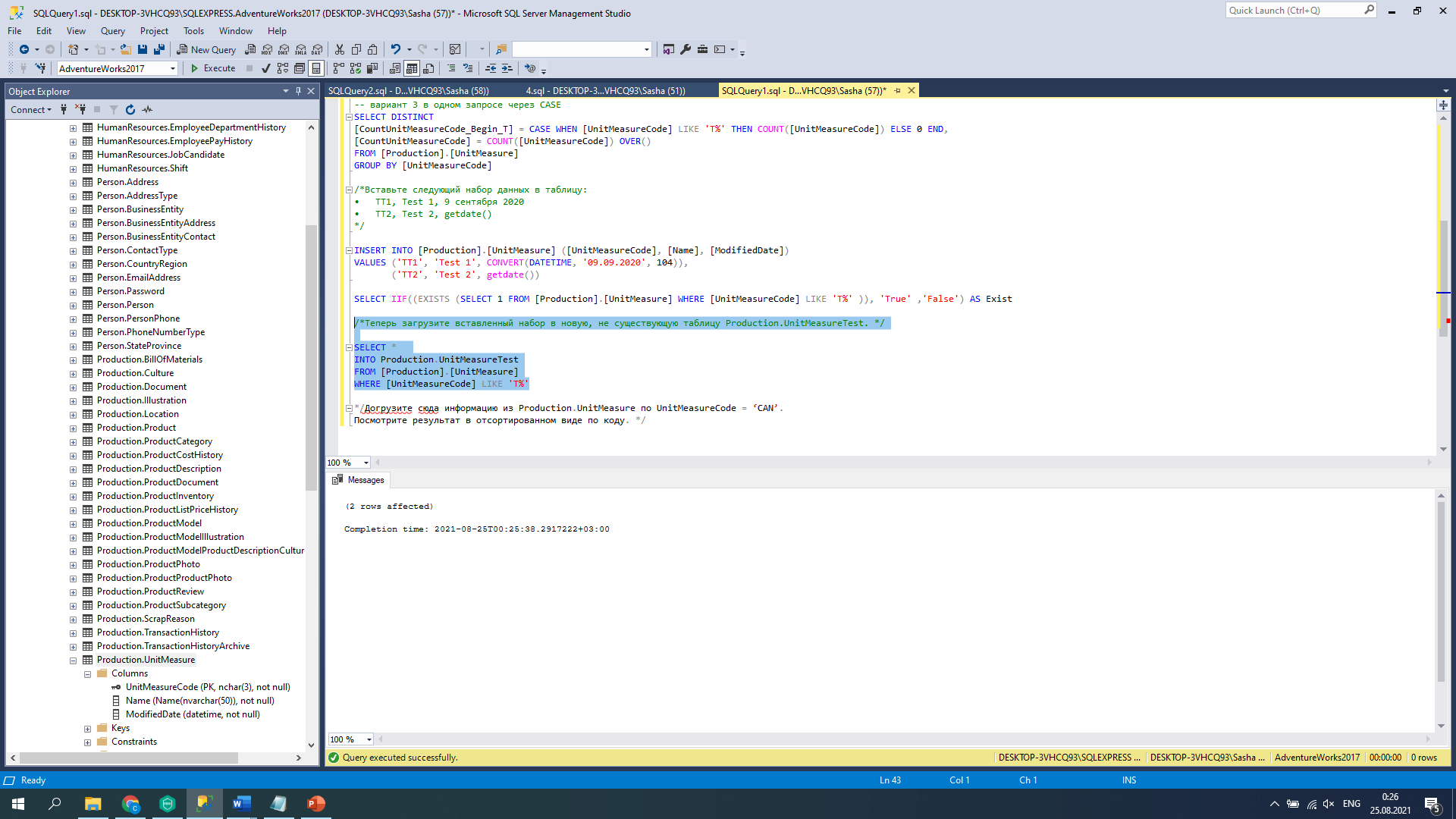
/\*Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. \*/

SELECT [UnitMeasureCode], [Name], [ModifiedDate]

INTO Production.UnitMeasureTest

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'



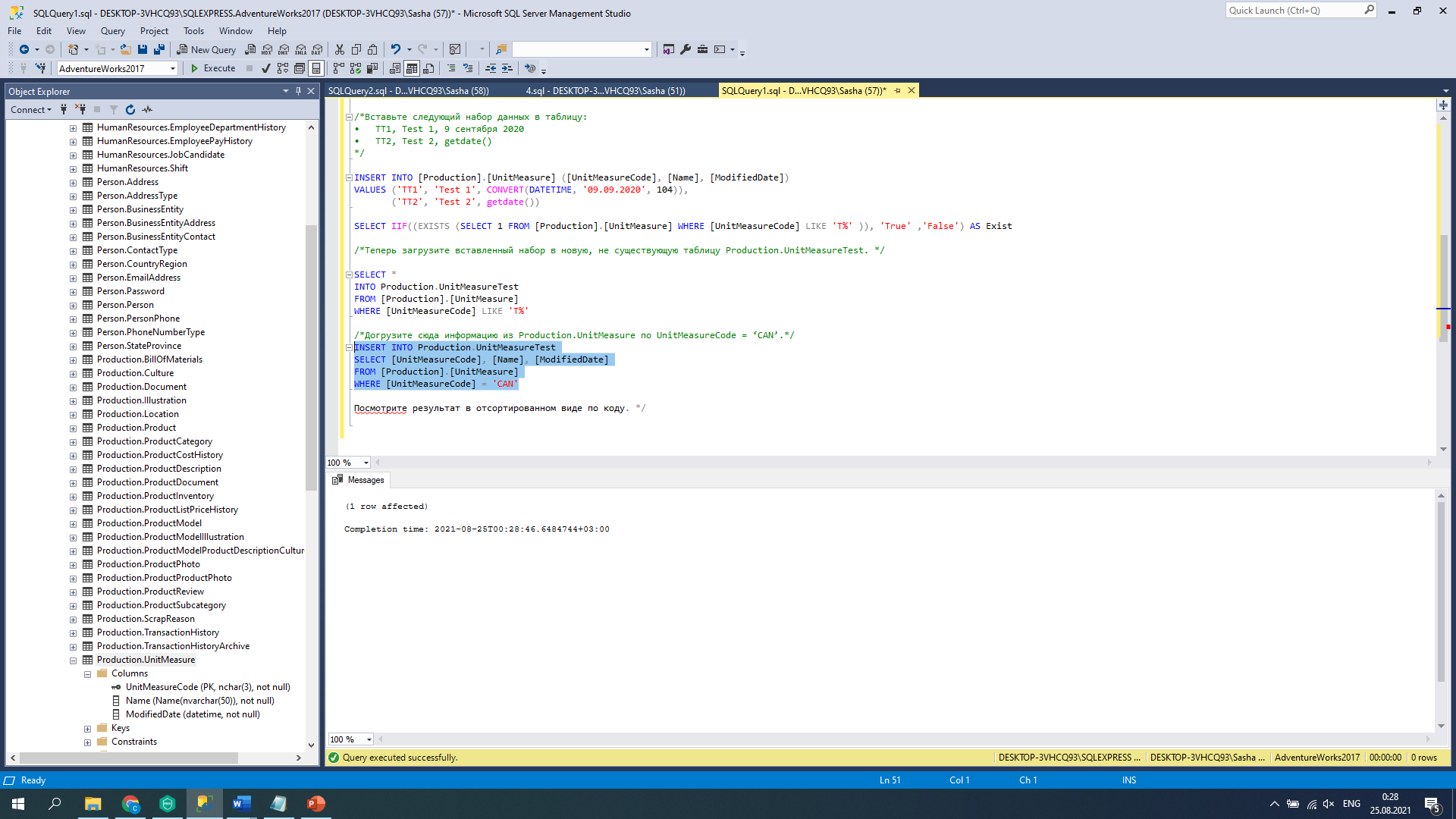
/\*Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’.\*/

INSERT INTO Production.UnitMeasureTest

SELECT [UnitMeasureCode], [Name], [ModifiedDate]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] = 'CAN'

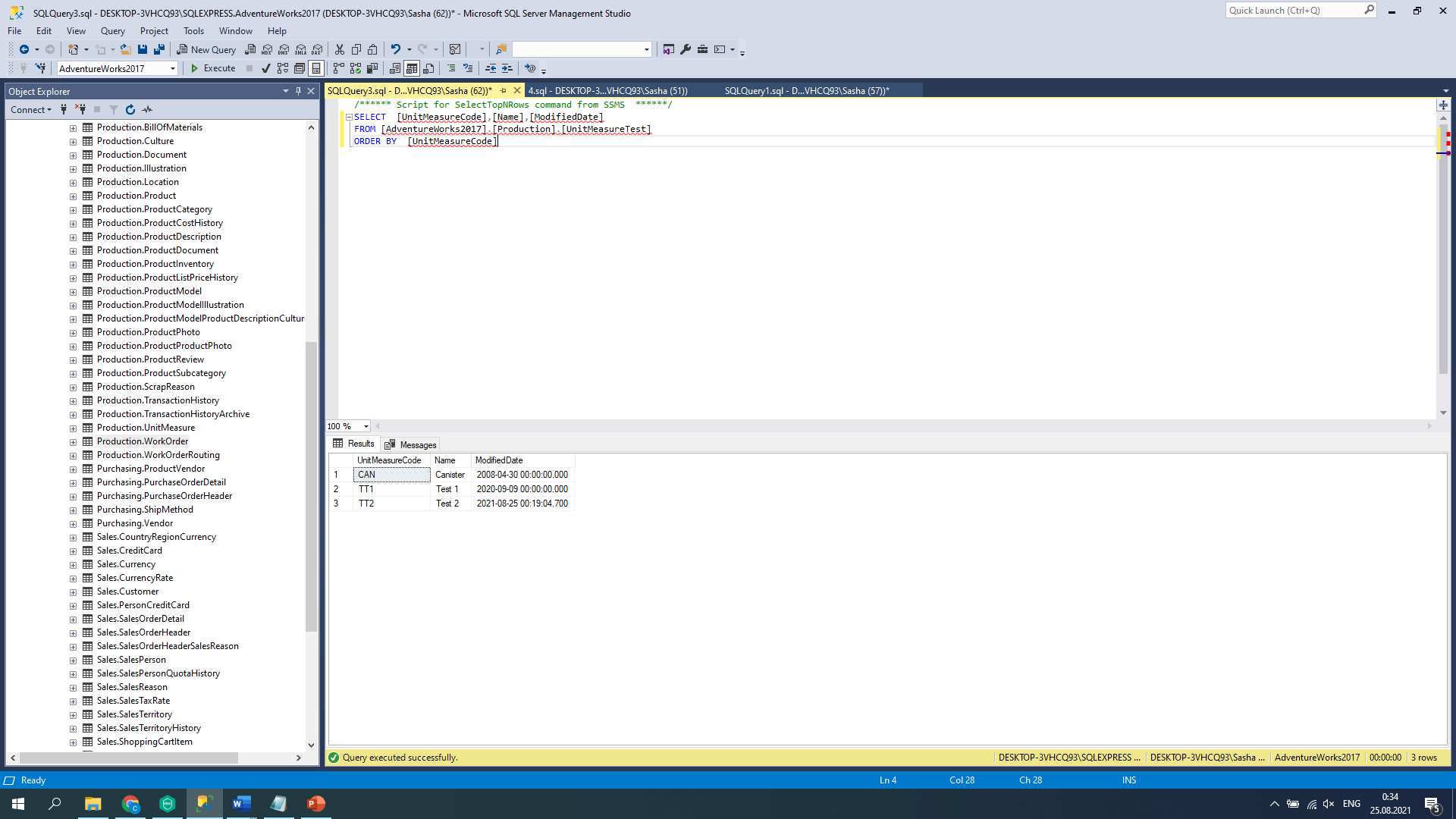


1. Посмотрите результат в отсортированном виде по коду.

SELECT [UnitMeasureCode],[Name],[ModifiedDate]

FROM [AdventureWorks2017].[Production].[UnitMeasureTest]

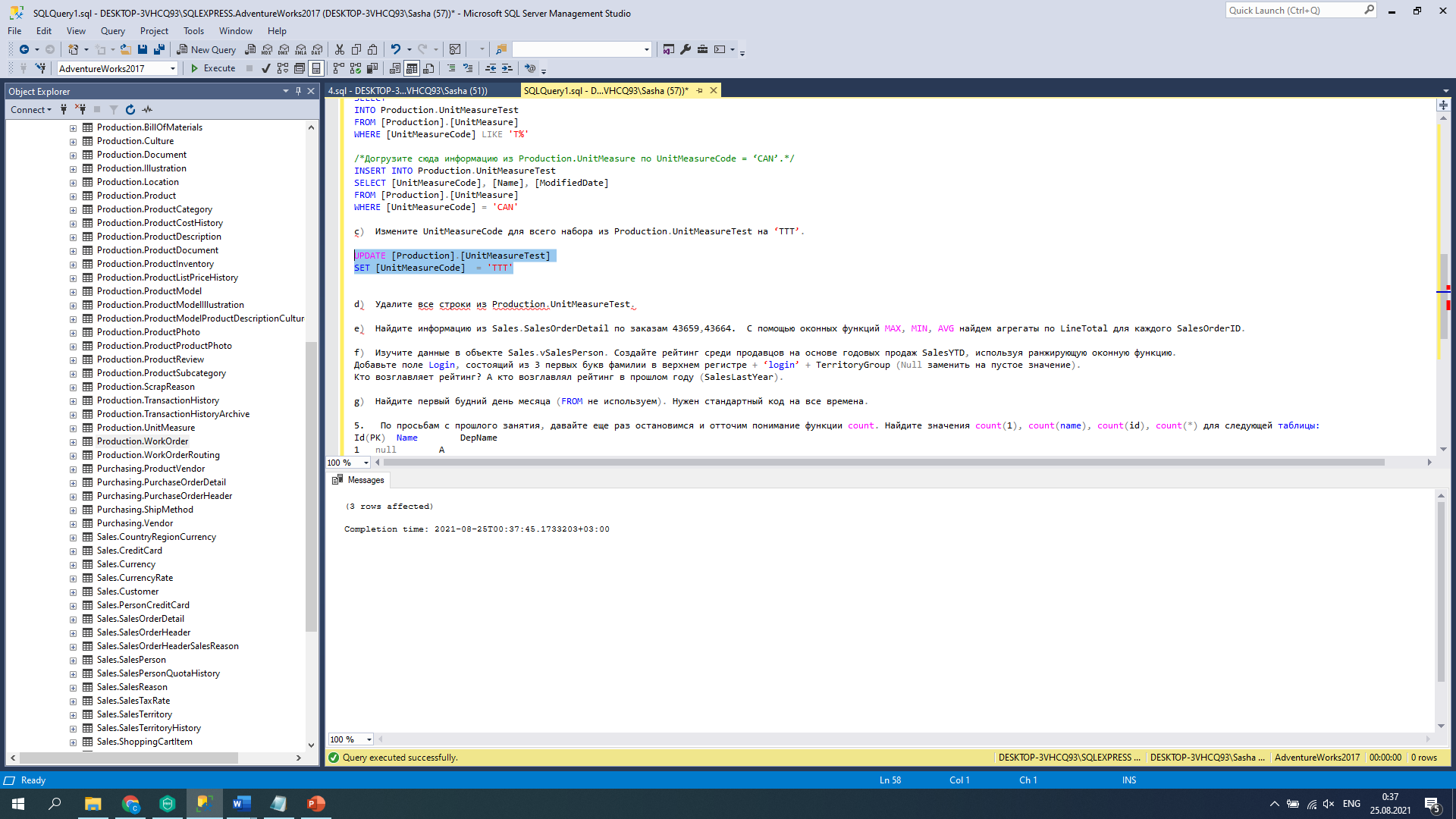
ORDER BY [UnitMeasureCode]



1. Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’.

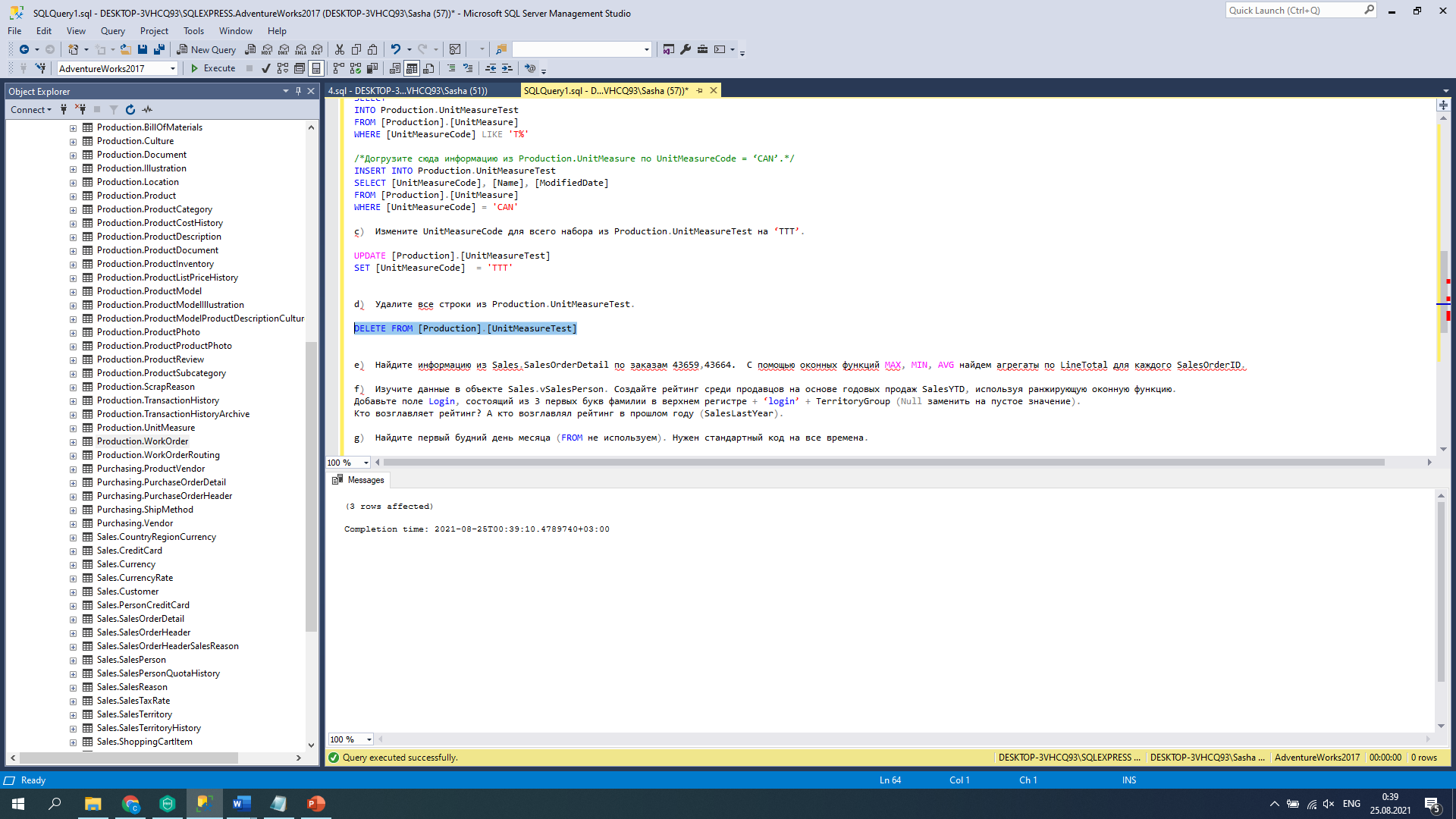
UPDATE [Production].[UnitMeasureTest]

SET [UnitMeasureCode] = 'TTT'



1. Удалите все строки из Production.UnitMeasureTest.

DELETE FROM [Production].[UnitMeasureTest]



1. Найдите информацию из Sales.SalesOrderDetail по заказам 43659,43664. С помощью оконных функций MAX, MIN, AVG найдем агрегаты по LineTotal для каждого SalesOrderID.

SELECT

SalesOrderID

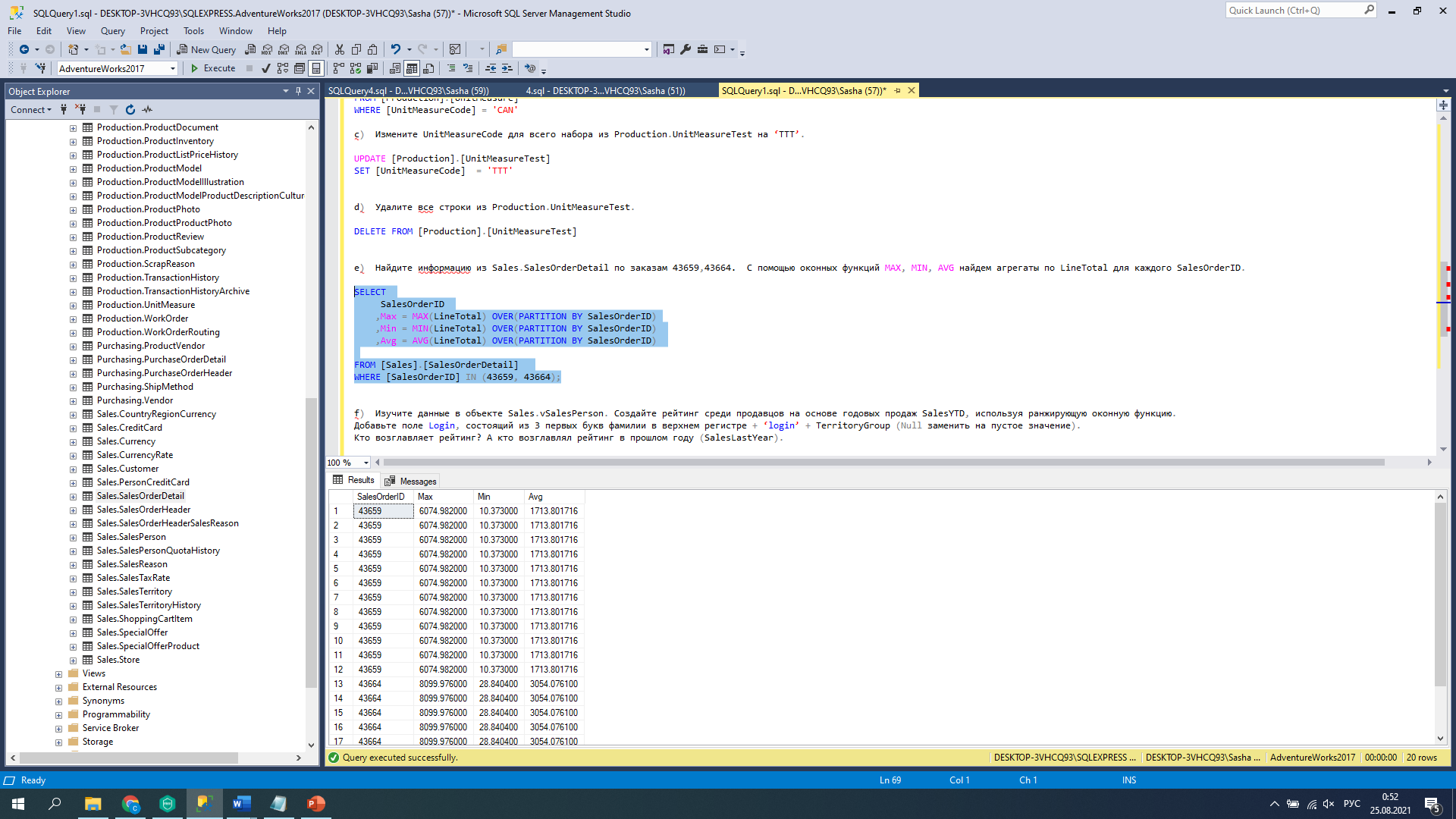
,Max = MAX(LineTotal) OVER(PARTITION BY SalesOrderID)

,Min = MIN(LineTotal) OVER(PARTITION BY SalesOrderID)

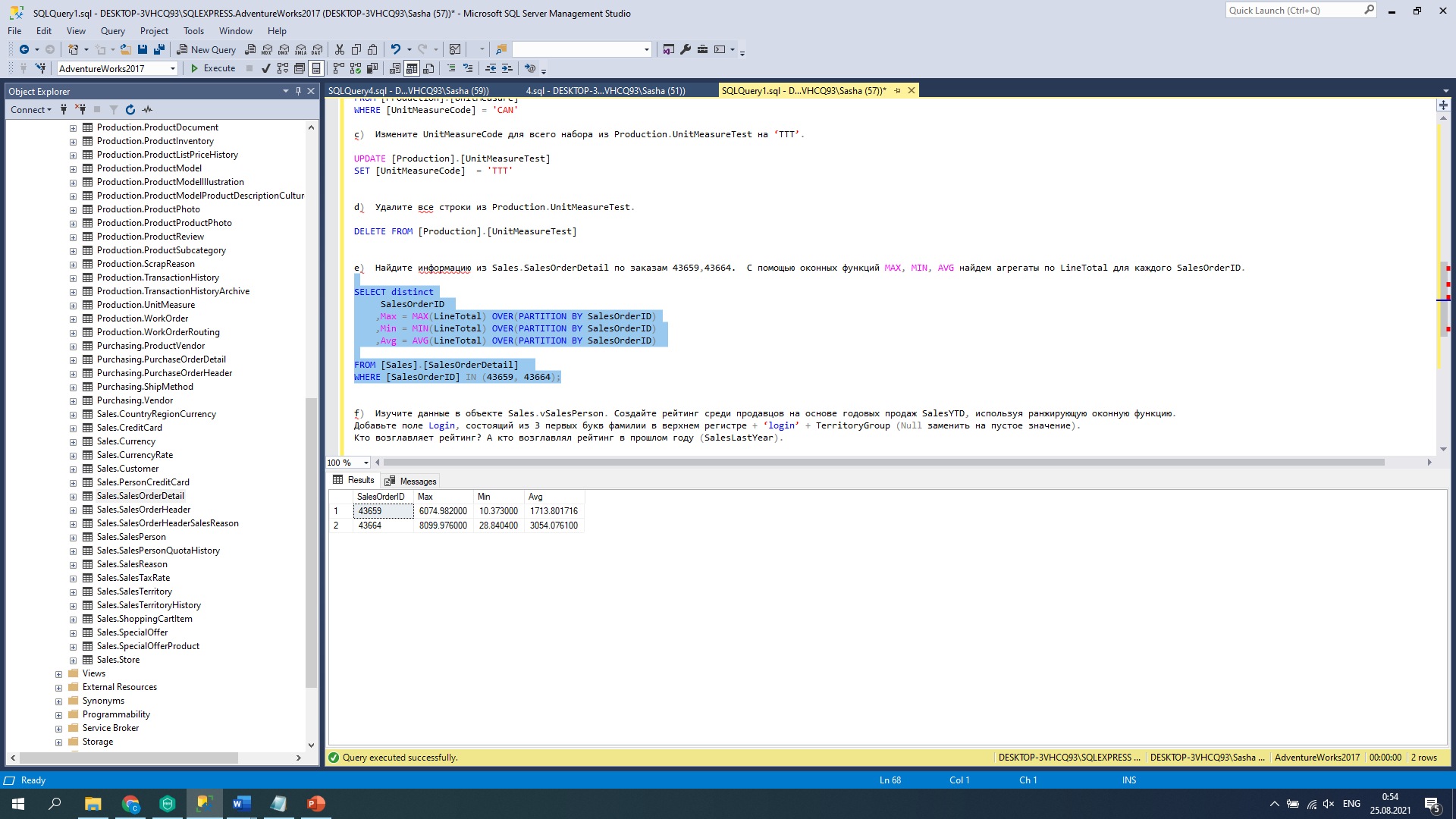
,Avg = AVG(LineTotal) OVER(PARTITION BY SalesOrderID)

FROM [Sales].[SalesOrderDetail]

WHERE [SalesOrderID] IN (43659, 43664);



Не поняла выражение «для каждого SalesOrderID» - для всех записей, у которых SalesOrderID , который мы находим, тогда как в ответе, если имеется ввиду только 2 SalesOrderID из условия, тогда использую Distinct



1. Изучите данные в объекте Sales.vSalesPerson. Создайте рейтинг cреди продавцов на основе годовых продаж SalesYTD, используя ранжирующую оконную функцию.

Добавьте поле Login, состоящий из 3 первых букв фамилии в верхнем регистре + ‘login’ + TerritoryGroup (Null заменить на пустое значение).

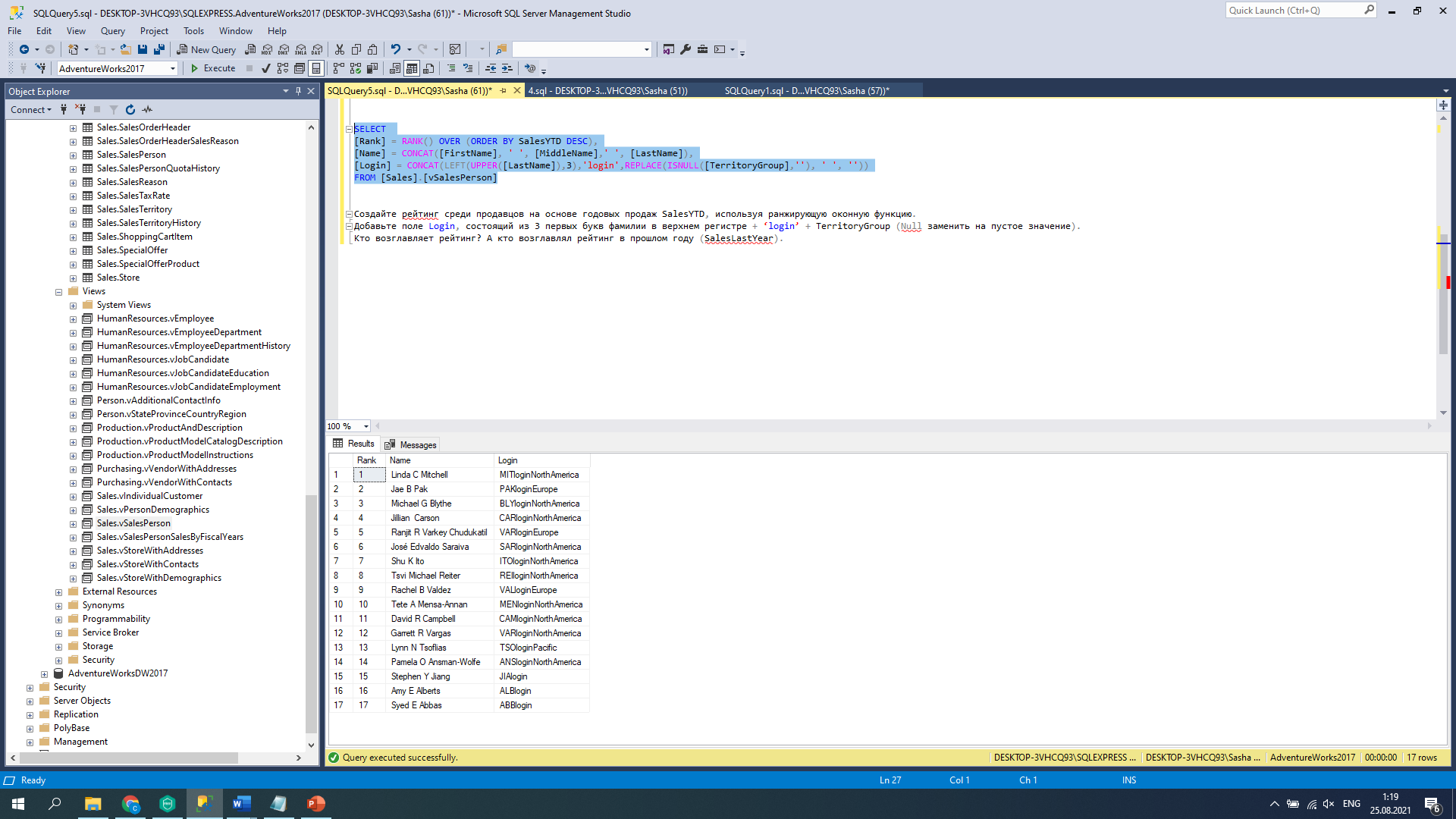
SELECT

[Rank] = RANK() OVER (ORDER BY SalesYTD DESC),

[Name] = CONCAT([FirstName], ' ', [MiddleName],' ', [LastName]),

[Login] = CONCAT(LEFT(UPPER([LastName]),3),'login',REPLACE(ISNULL([TerritoryGroup],''), ' ', ''))

FROM [Sales].[vSalesPerson]



Кто возглавляет рейтинг? А кто возглавлял рейтинг в прошлом году (SalesLastYear).

SELECT DISTINCT

[LeadCurrentYear] = FIRST\_VALUE([Name]) OVER (ORDER BY [Rank]),

[LeadLastYear] = FIRST\_VALUE([Name]) OVER (ORDER BY [RankLastYear])

FROM (

SELECT

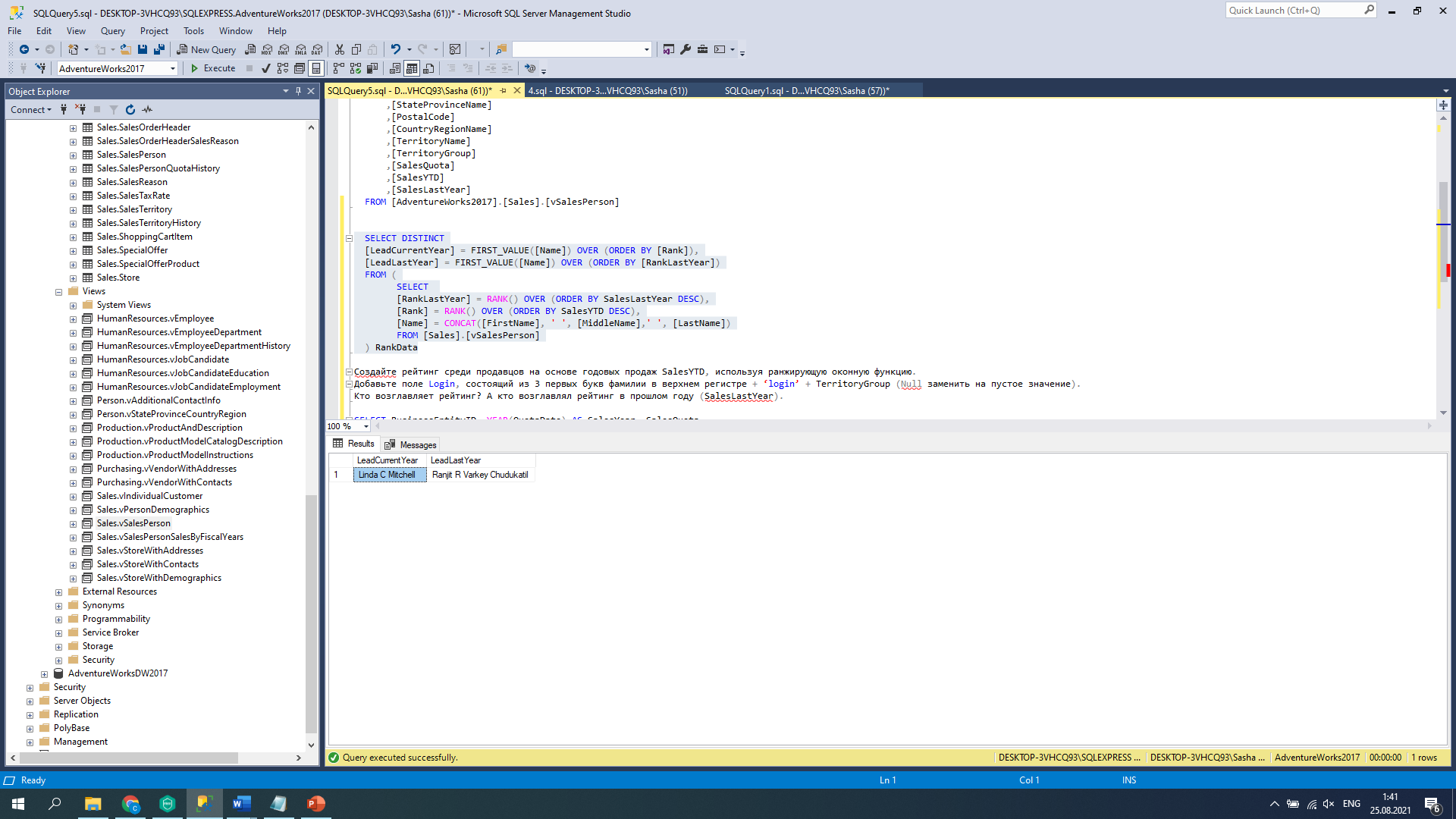
[RankLastYear] = RANK() OVER (ORDER BY SalesLastYear DESC),

[Rank] = RANK() OVER (ORDER BY SalesYTD DESC),

[Name] = CONCAT([FirstName], ' ', [MiddleName],' ', [LastName])

FROM [Sales].[vSalesPerson]

) RankData



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

SELECT FirstWorkDay =

CASE DATEPART(WEEKDAY, DATEADD(month, DATEDIFF(month, 0, GETDATE()), 0))

WHEN 6 THEN CONVERT(NVARCHAR, DATEADD(month, DATEDIFF(month, 0, GETDATE()), 2), 104)

WHEN 7 THEN CONVERT(NVARCHAR, DATEADD(month, DATEDIFF(month, 0, GETDATE()), 1), 104)

ELSE CONVERT(NVARCHAR,DATEADD(month, DATEDIFF(month, 0, GETDATE()), 0),104) END

Для расчёта на любой месяц GETDATE() меняем на нужную дату:

1. Рассчитывает 1й день месяца указанной даты
2. Проверяет 1е число какой день недели
3. Если 6, к первому числу прибавляет 2 дня
4. Если 7 к первому числу месяца прибавляет 1 день
5. Иначе первый день месяца и является первым будним днем
6. По просьбам с прошлого занятия, давайте еще раз остановимся и отточим понимание функции count. Найдите значения count(1), count(name), count(id), count(\*) для следующей таблицы:

**Id**(PK) **Name DepName**

1 null A

2 null null

3 A C

4 B C

|  |  |
| --- | --- |
| count(1) | 4 |
| count(name) | 2 |
| count(id) | 4 |
| count(\*) | 4 |