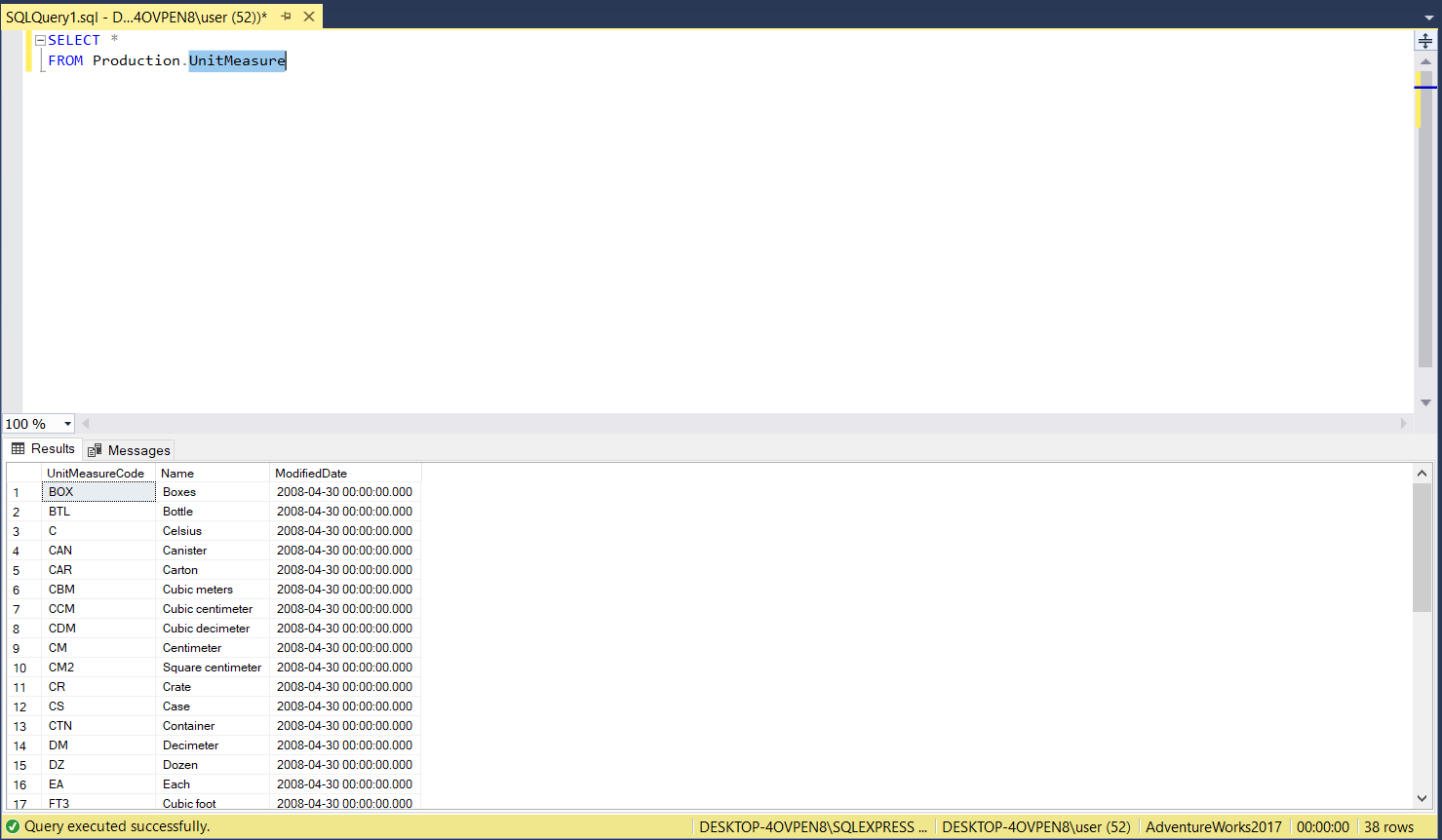
1. **При каких значениях оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат?**

Если не будет повторяющихся значений в столбце по которому делаем ORDER BY

1. **Решите на базе данных AdventureWorks2017 следующие задачи.**
2. **Изучите данные в таблице Production.UnitMeasure.**

SELECT \*

FROM Production.UnitMeasure

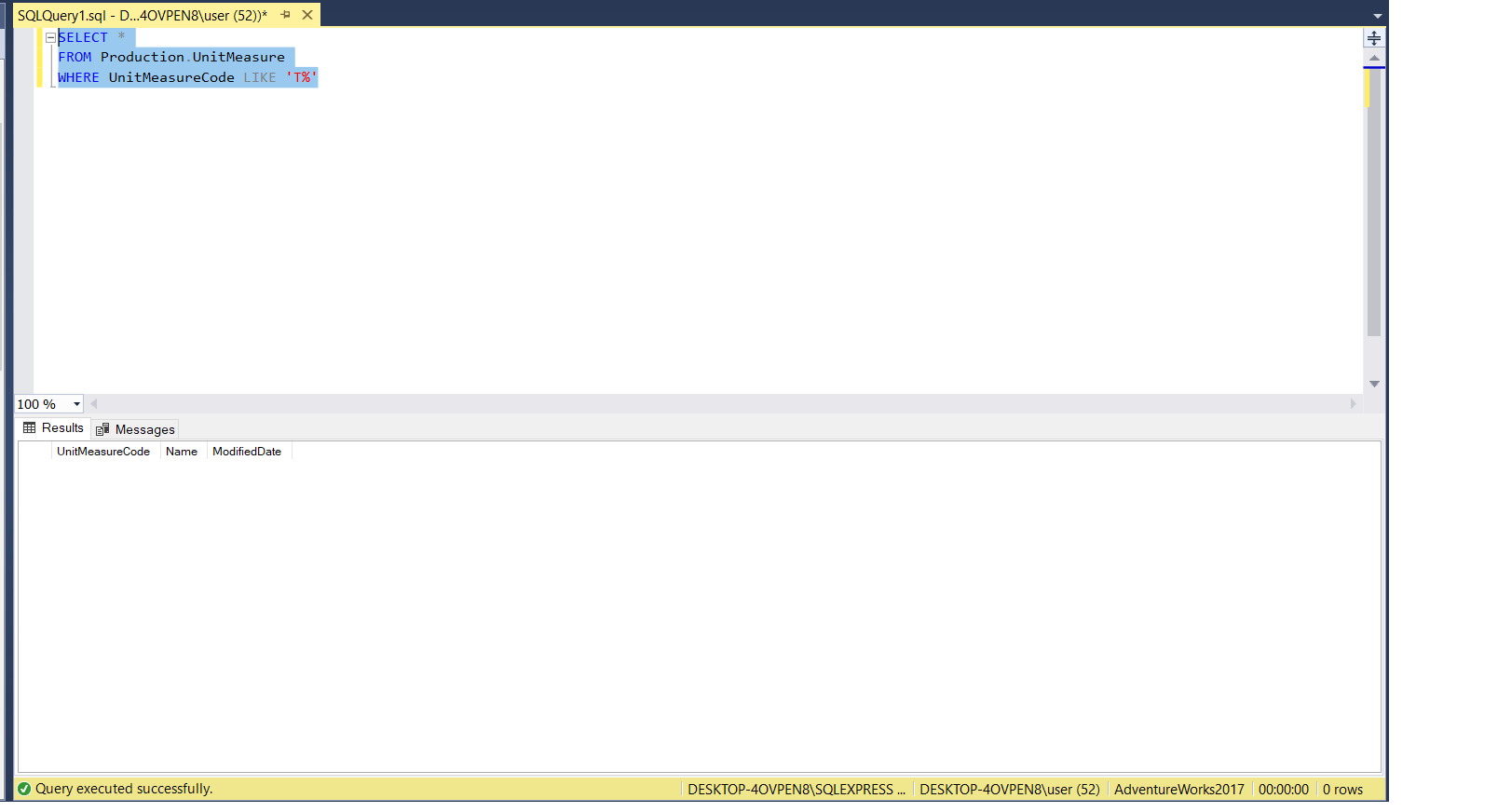


**Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’.**

SELECT \*

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'



**Сколько всего различных кодов здесь есть?**

Всего в таблице 38 кодов, начинающихся на ‘T’ – 0

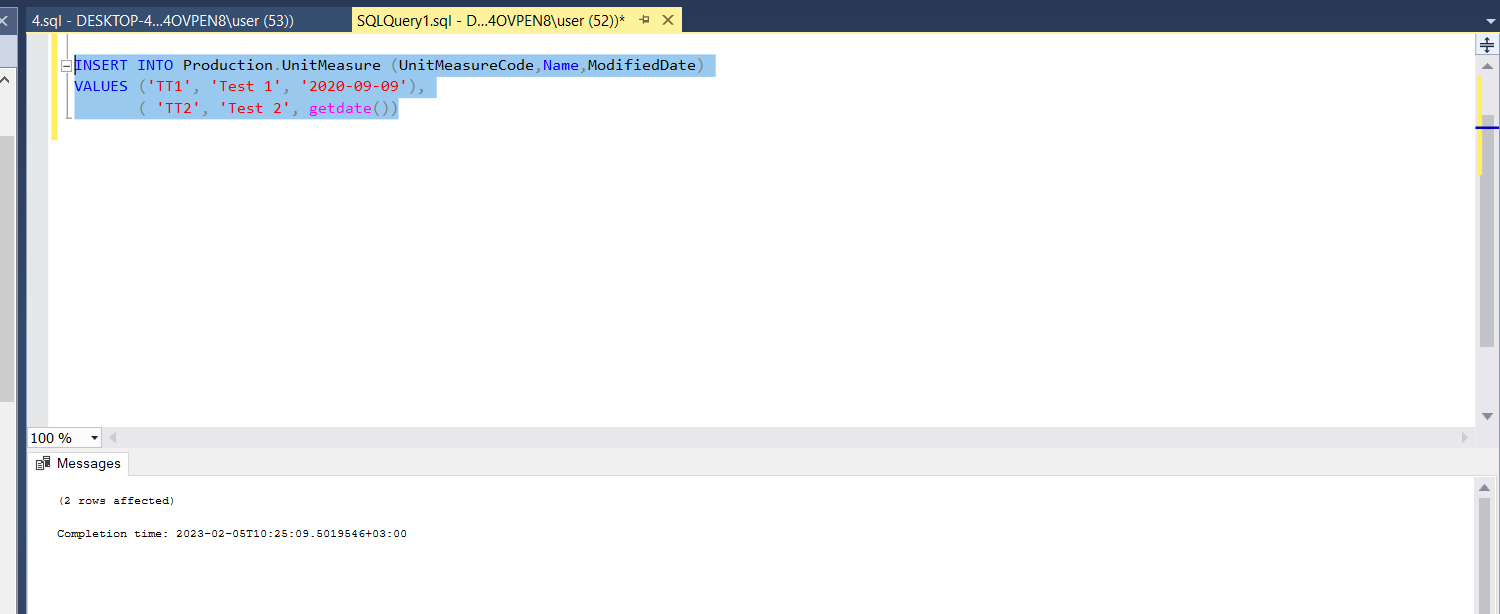
**Вставьте следующий набор данных в таблицу:**

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

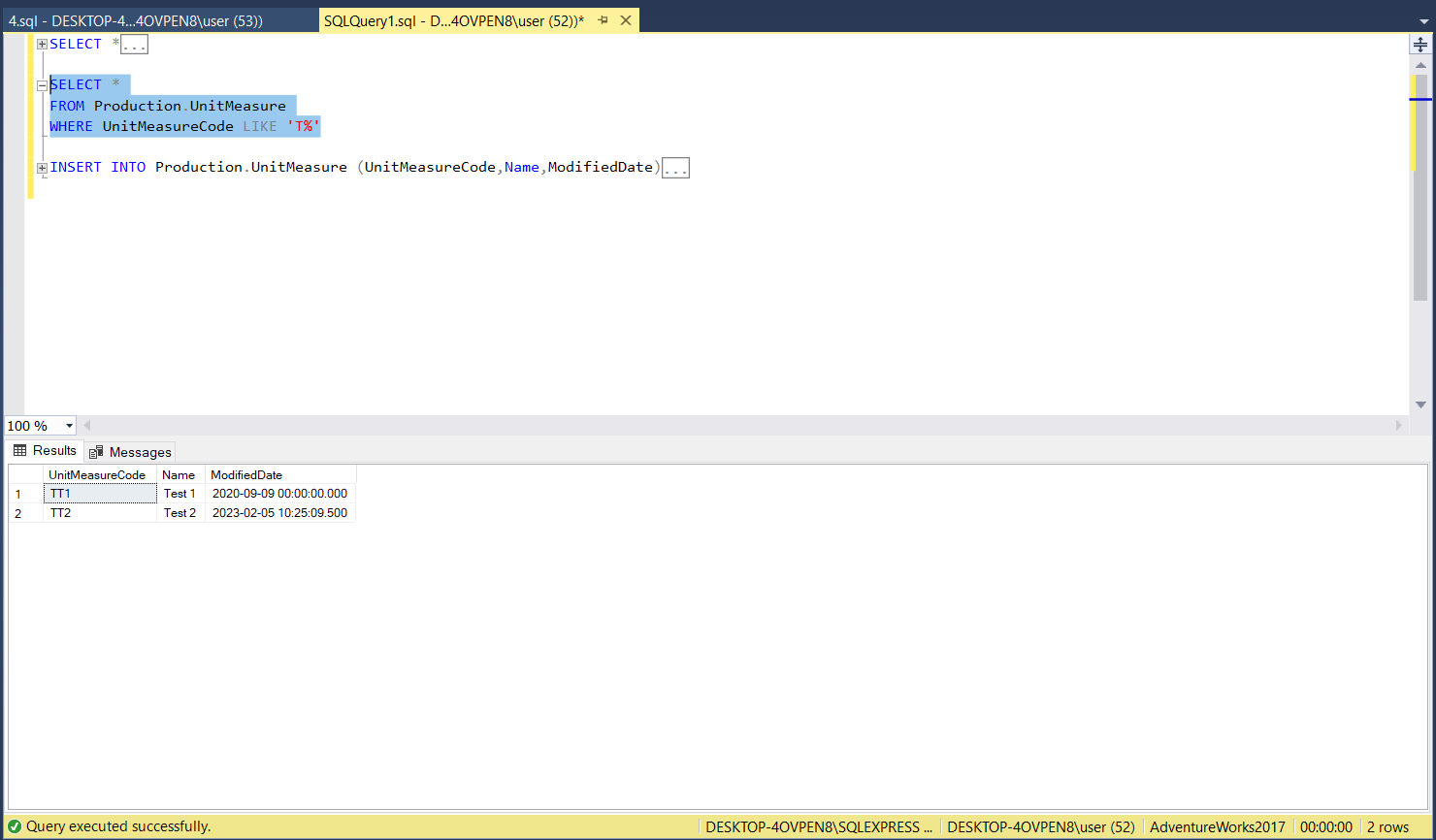
INSERT INTO Production.UnitMeasure (UnitMeasureCode,Name,ModifiedDate)

VALUES ('TT1', 'Test 1', '2020-09-09'),

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



**Проверьте теперь, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. -** 2 СТРОКИ



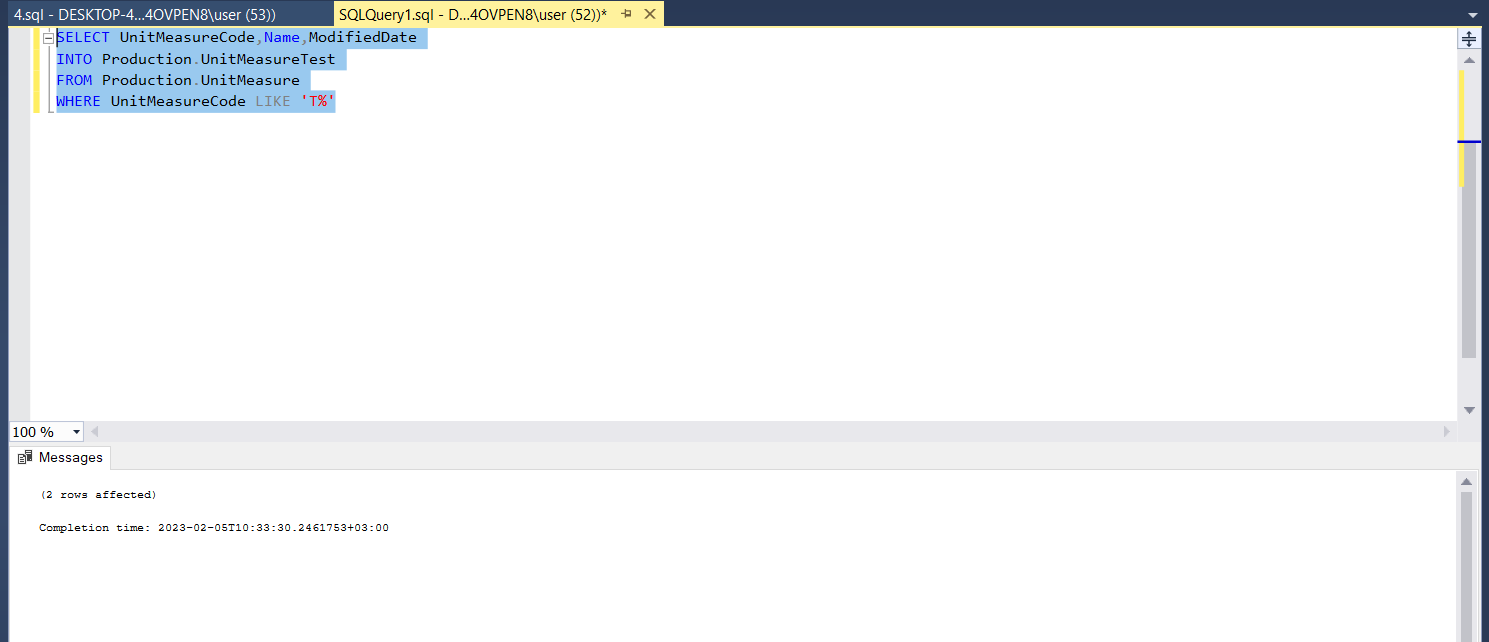
1. **Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest.**

SELECT UnitMeasureCode, Name, ModifiedDate

INTO Production.UnitMeasureTest

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'



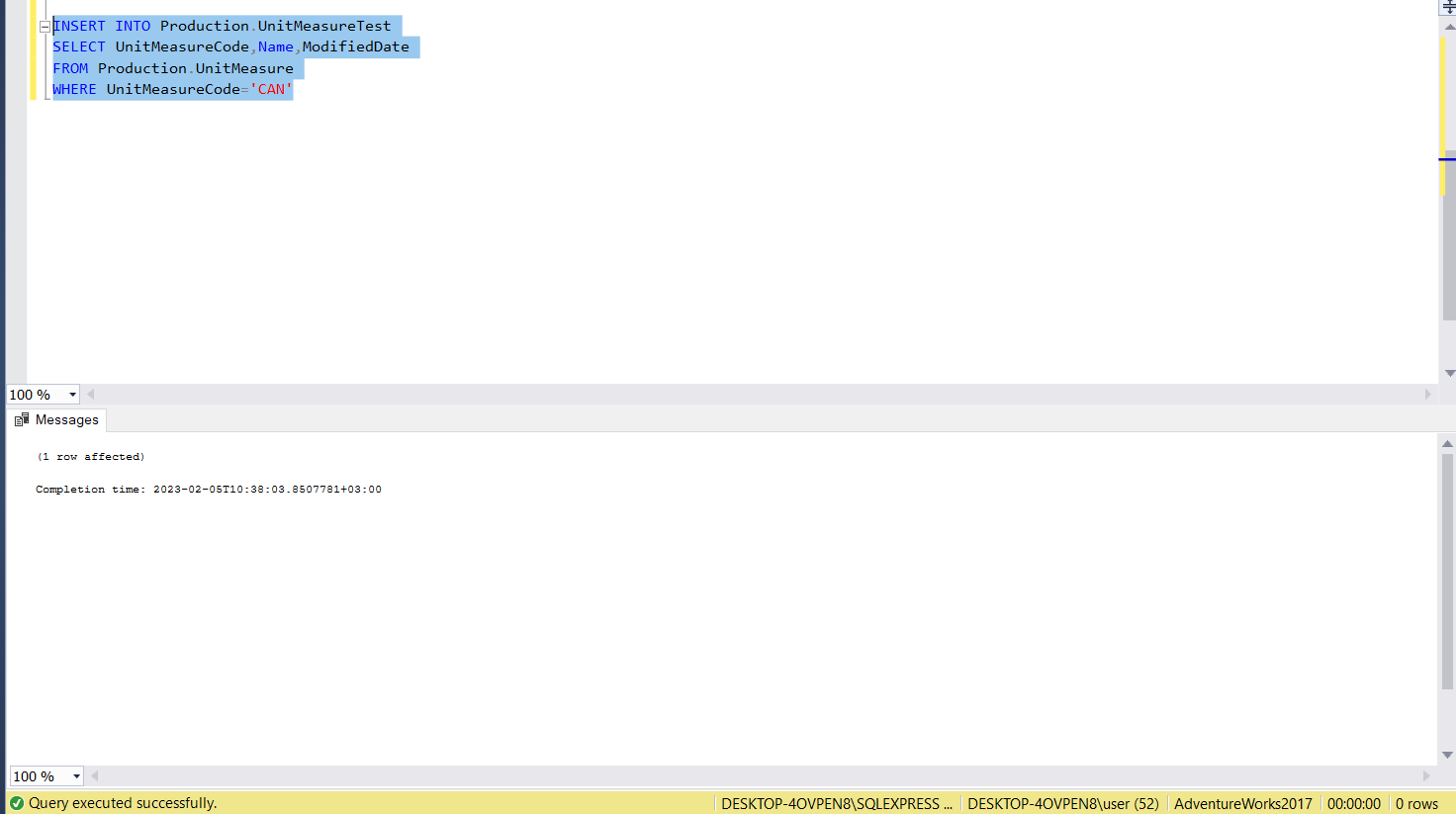
**Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’.**

INSERT INTO Production.UnitMeasureTest

SELECT UnitMeasureCode,Name,ModifiedDate

FROM Production.UnitMeasure

WHERE UnitMeasureCode='CAN'

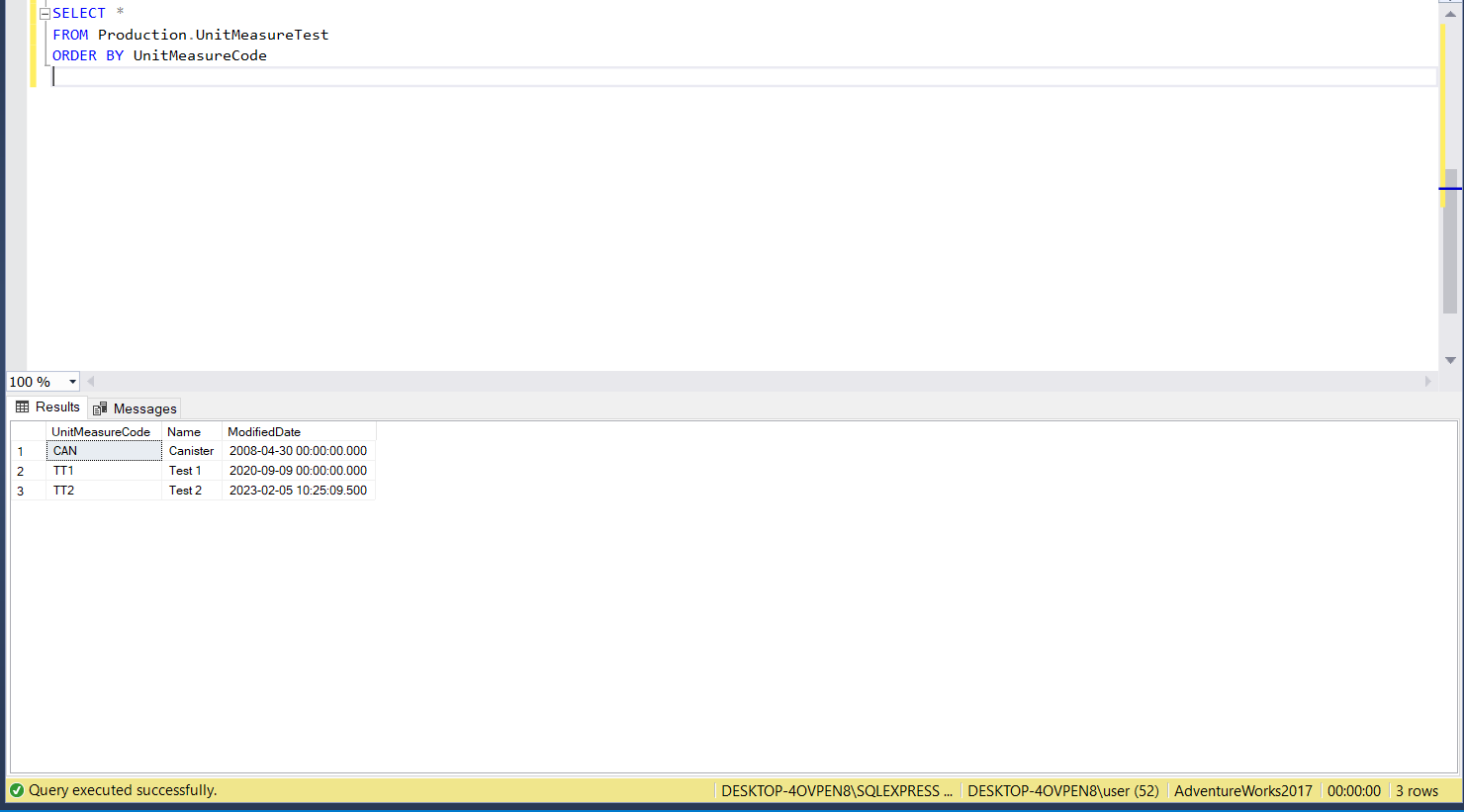


**Посмотрите результат в отсортированном виде по коду.**

SELECT \*

FROM Production.UnitMeasureTest

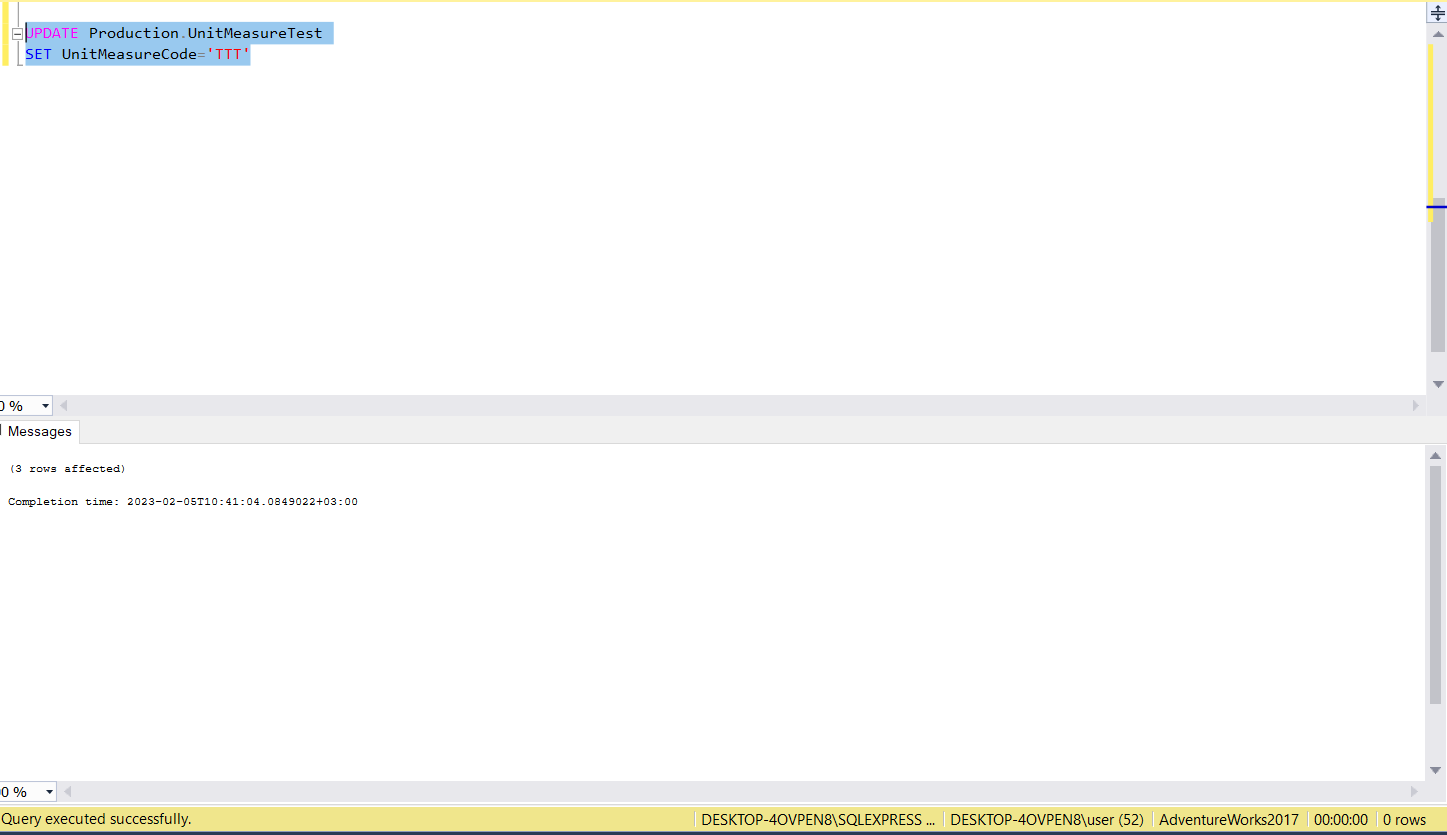
ORDER BY UnitMeasureCode



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

UPDATE Production.UnitMeasureTest

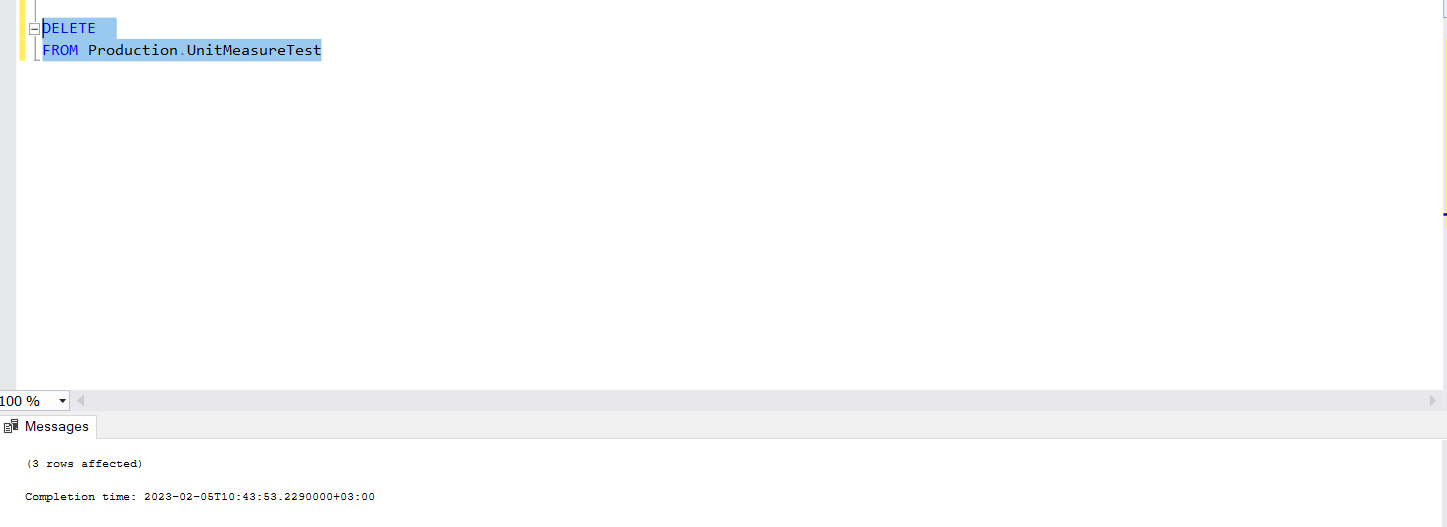
SET UnitMeasureCode=’TTT’



1. **Удалите все строки из Production.UnitMeasureTest.**

DELETE

FROM Production.UnitMeasureTest

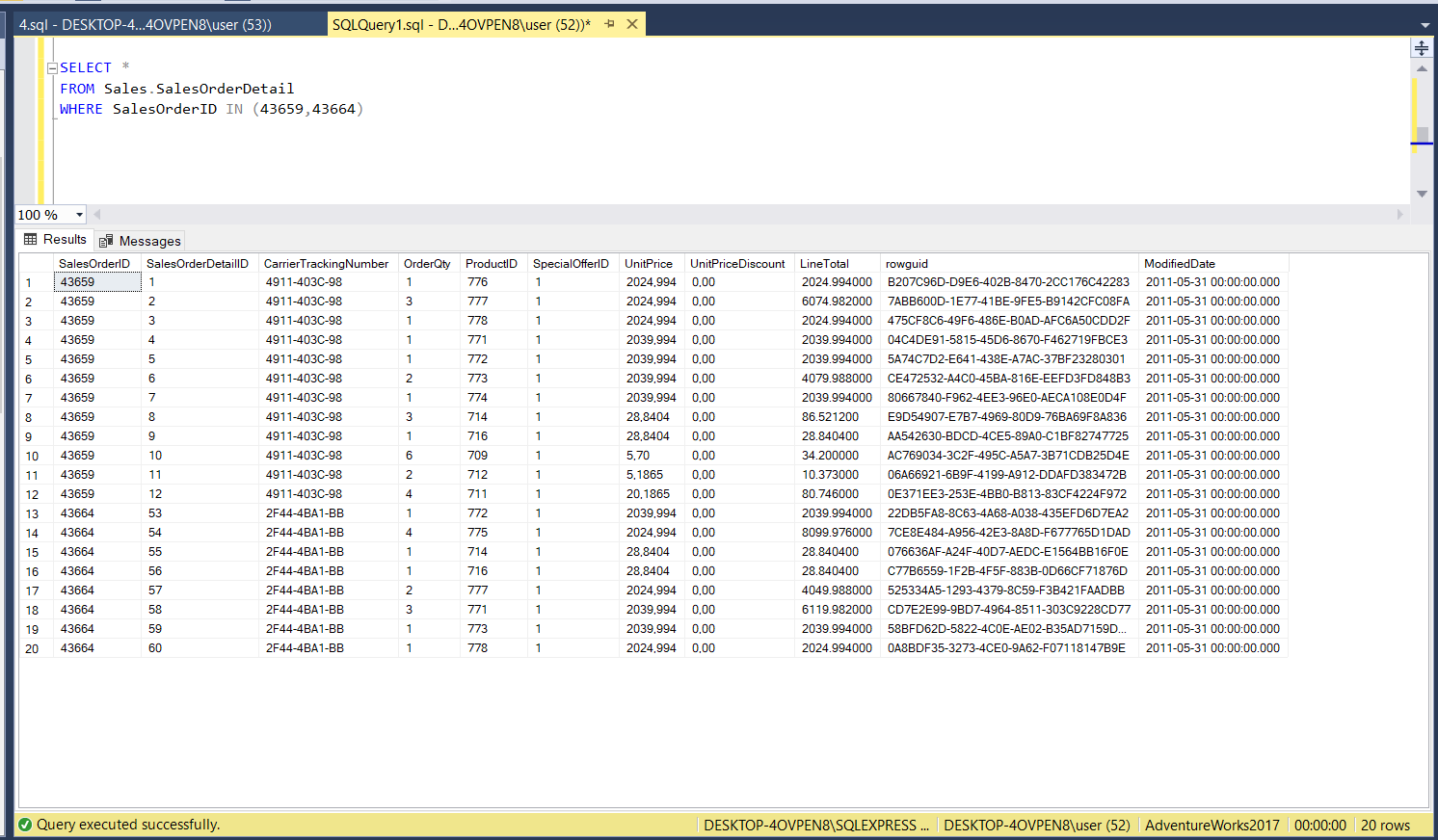


1. **Найдите информацию из Sales.SalesOrderDetail по заказам 43659,43664.**

SELECT \*

FROM Sales.SalesOrderDetail

WHERE SalesOrderID IN (43659,43664)



**С помощью оконных функций MAX, MIN, AVG найдем агрегаты по LineTotal для каждого SalesOrderID.**

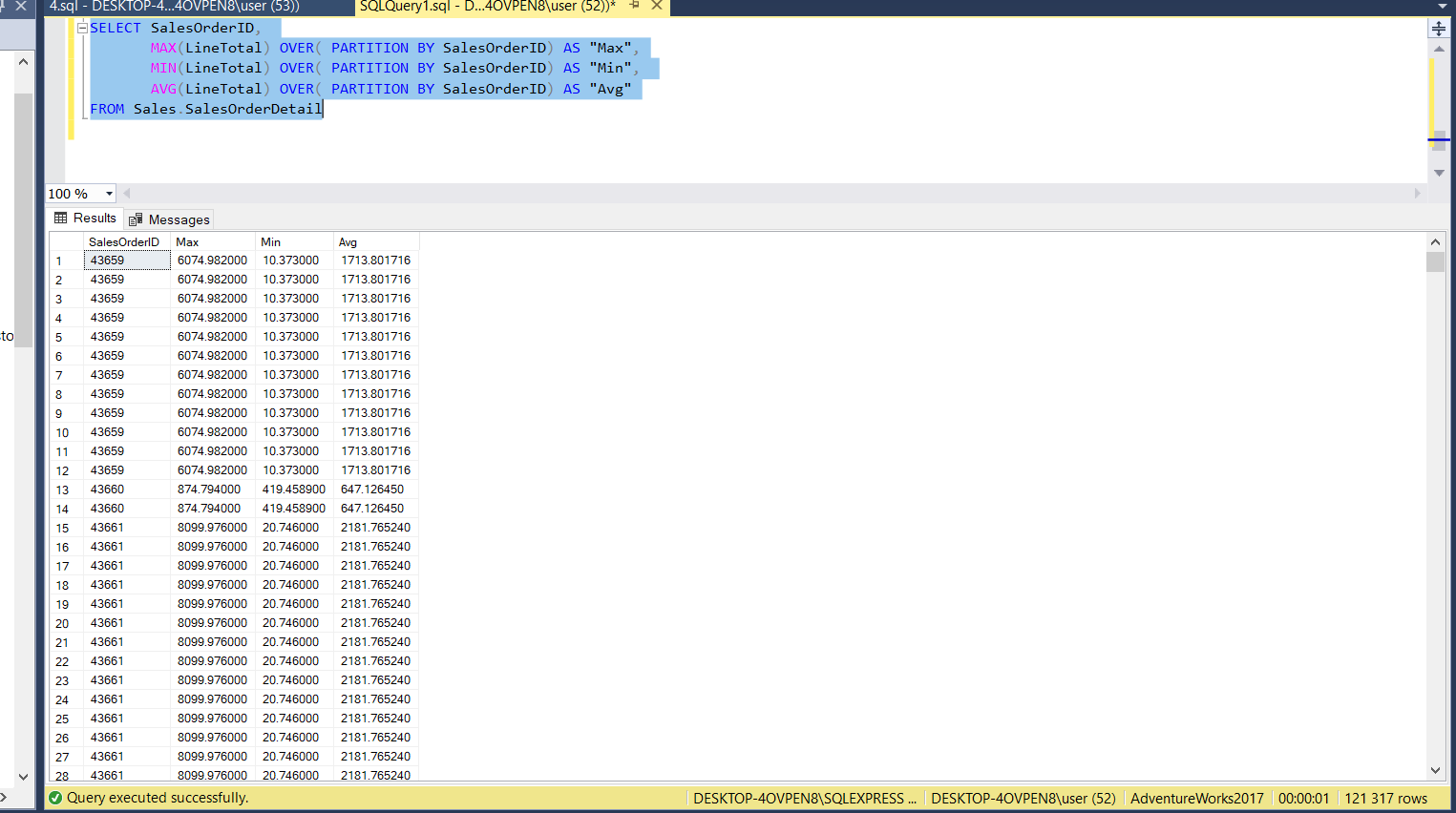
SELECT SalesOrderID,

MAX(LineTotal) OVER( PARTITION BY SalesOrderID) AS "Max",

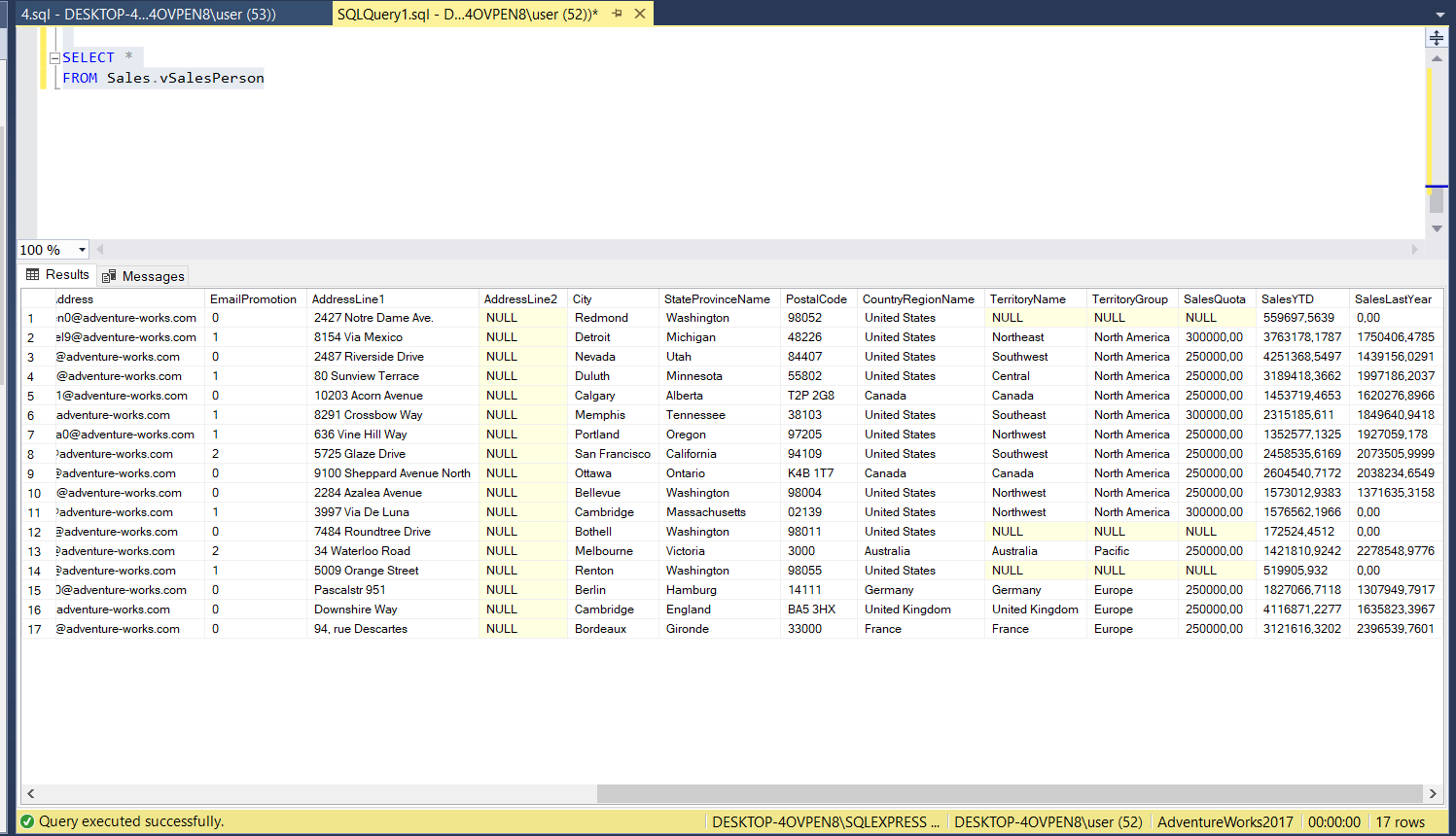
MIN(LineTotal) OVER( PARTITION BY SalesOrderID) AS "Min",

AVG(LineTotal) OVER( PARTITION BY SalesOrderID) AS "Avg"

FROM Sales.SalesOrderDetail



1. **Изучите данные в объекте Sales.vSalesPerson.**



**Создайте рейтинг cреди продавцов на основе годовых продаж SalesYTD, используя ранжирующую оконную функцию.**

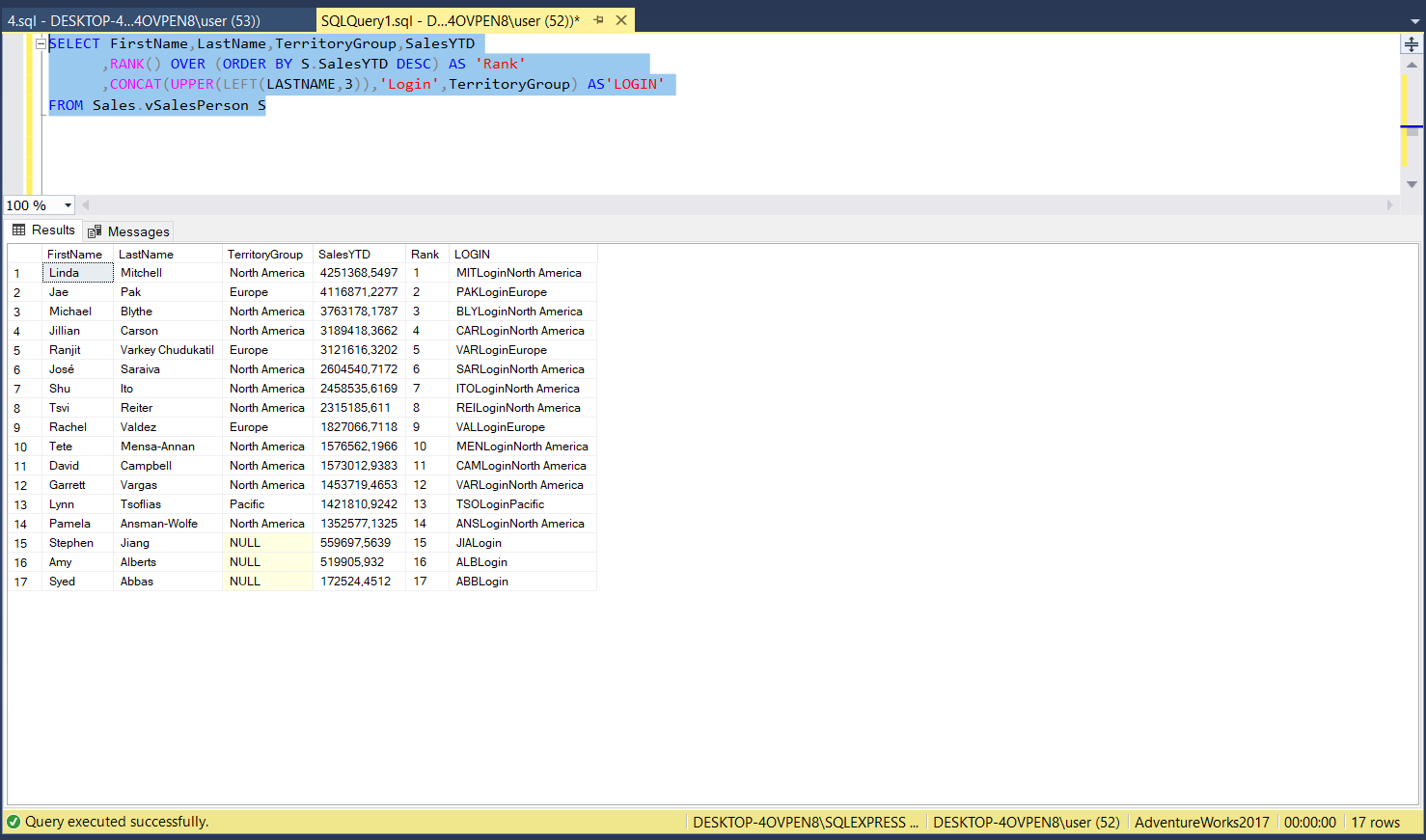
**Добавьте поле Login, состоящий из 3 первых букв фамилии в верхнем регистре + ‘login’ + TerritoryGroup (Null заменить на пустое значение).**

SELECT FirstName,LastName,TerritoryGroup,SalesYTD

,RANK() OVER (ORDER BY S.SalesYTD DESC) AS 'Rank'

,CONCAT(UPPER(LEFT(LASTNAME,3)),'Login',TerritoryGroup) AS'LOGIN'

FROM Sales.vSalesPerson S



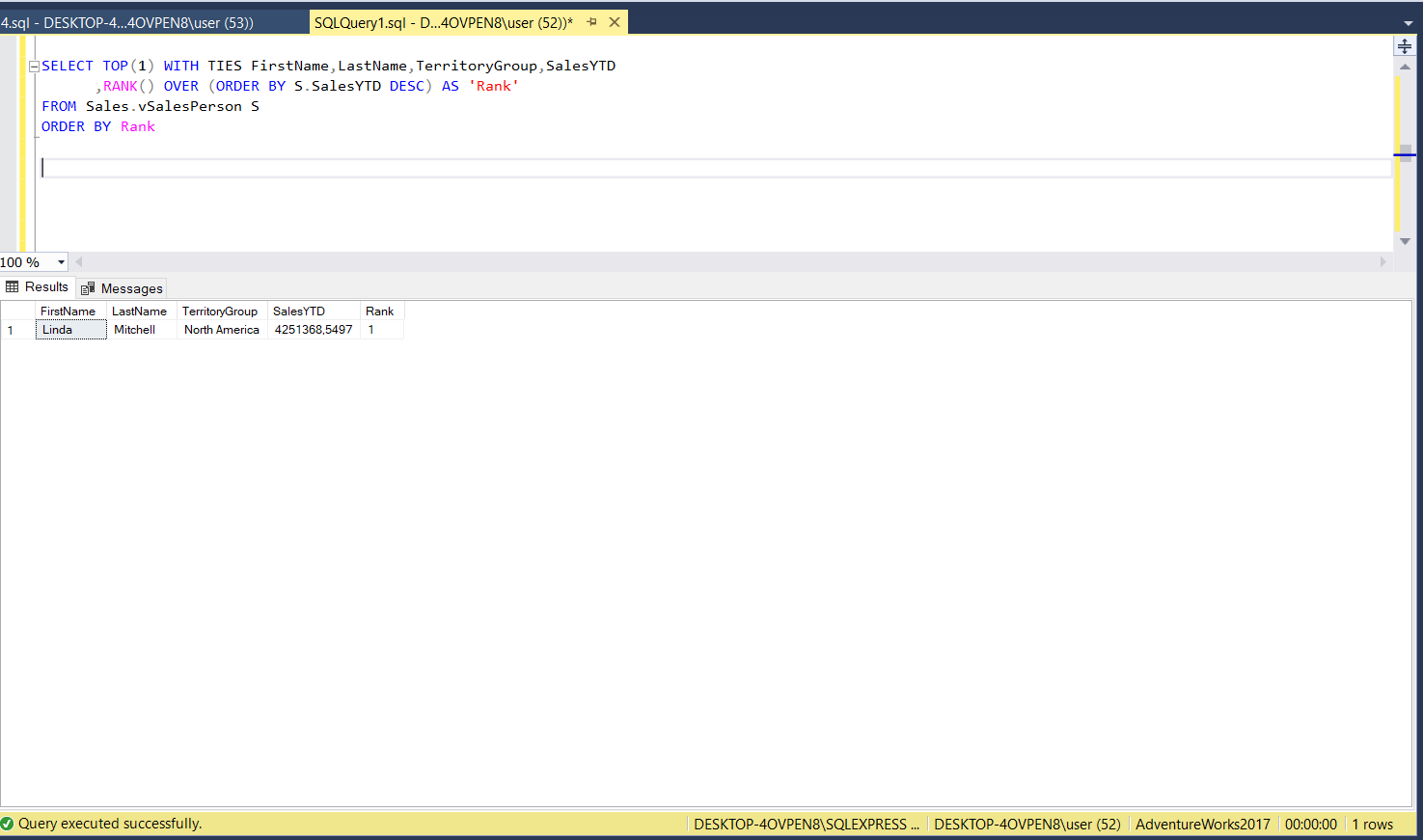
Кто возглавляет рейтинг?

SELECT TOP(1) WITH TIES FirstName,LastName,TerritoryGroup,SalesYTD

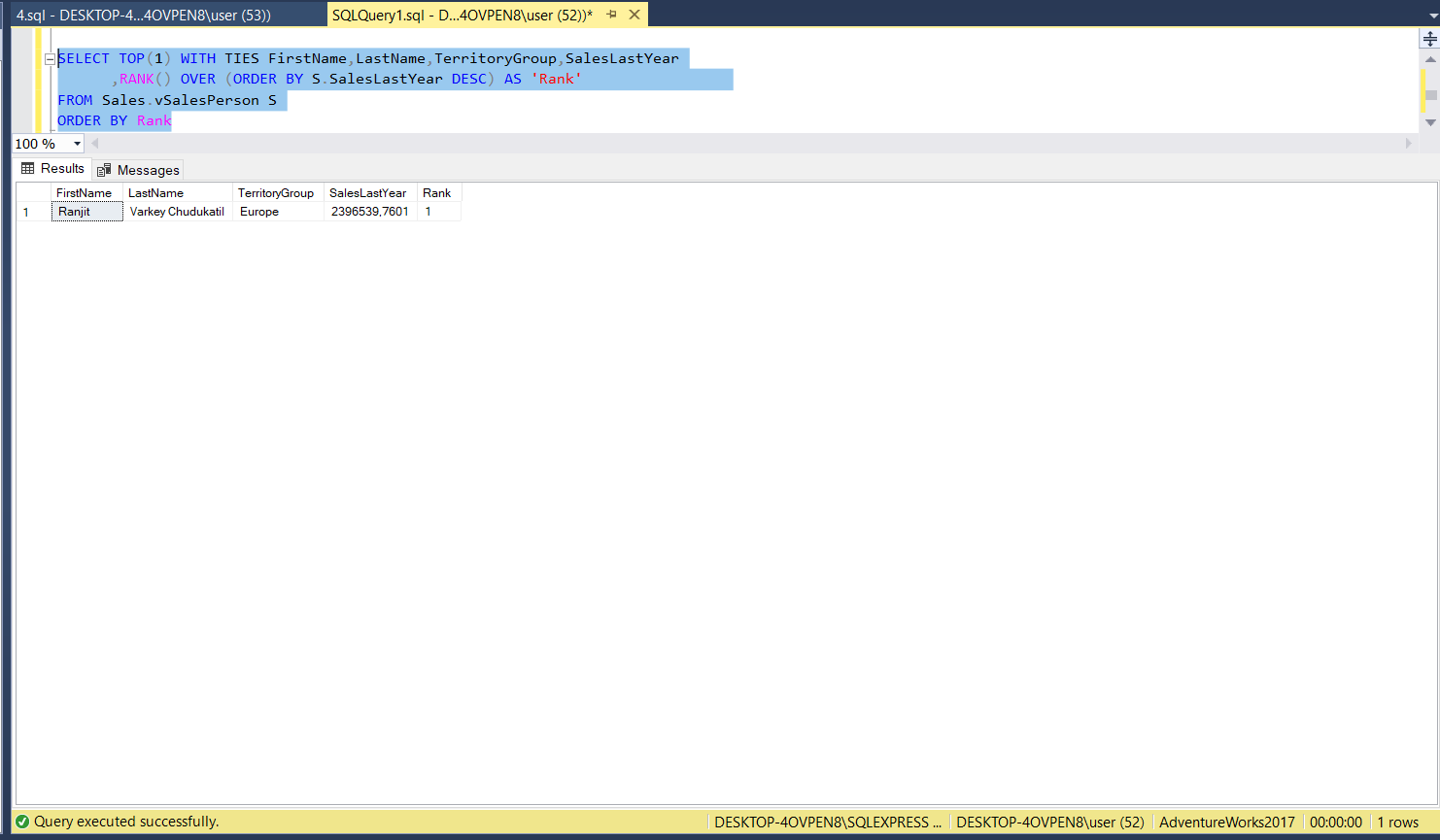
,RANK() OVER (ORDER BY S.SalesYTD DESC) AS 'Rank'

FROM Sales.vSalesPerson S

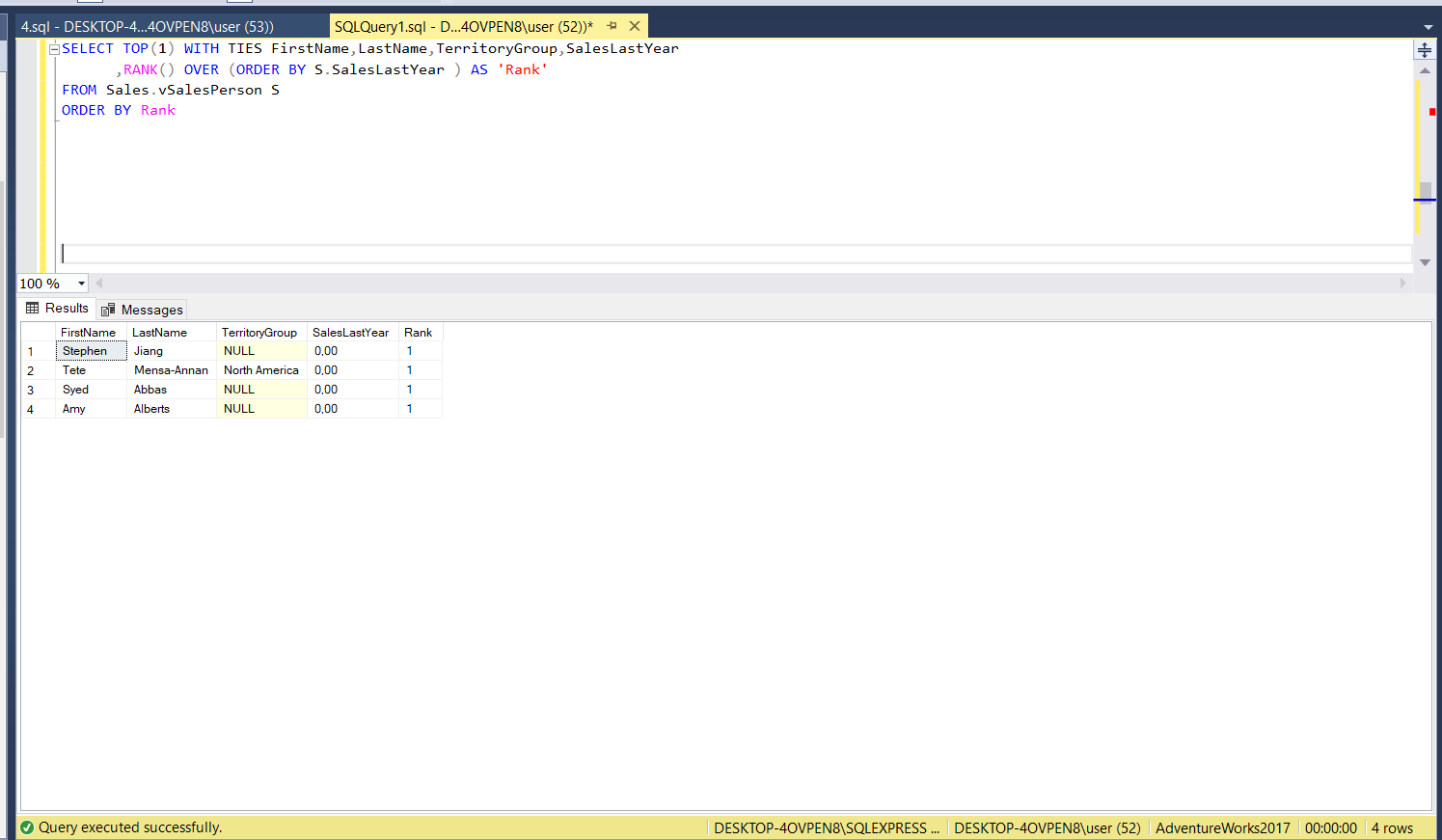
ORDER BY Rank



А кто возглавлял рейтинг в прошлом году (SalesLastYear).



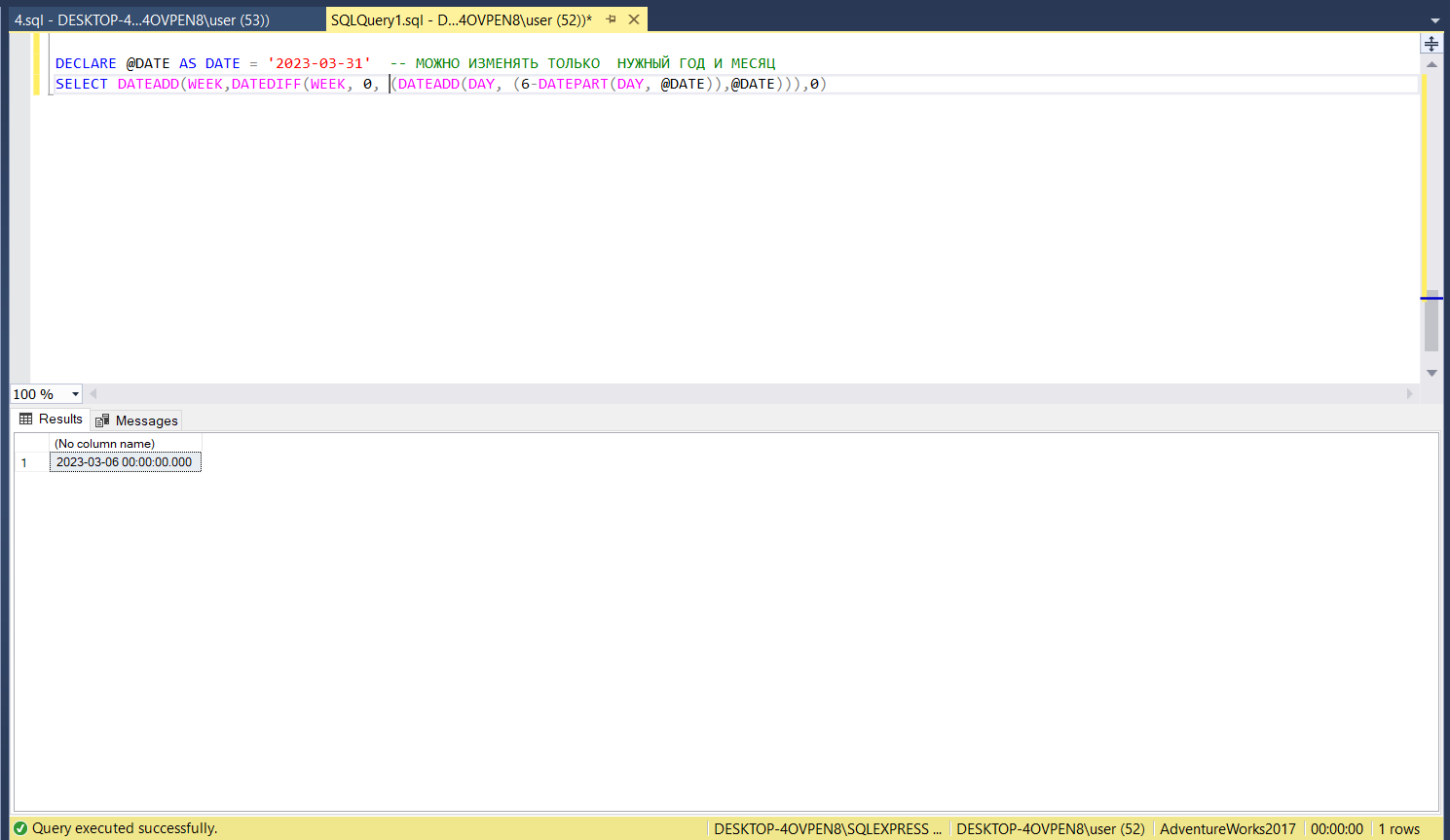
БЕЗ desc- кто хуже



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

DECLARE @DATE AS DATE = '2023-03-01'

SELECT DATEADD(WEEK,DATEDIFF(WEEK, 0, (DATEADD(DAY, (6-DATEPART(DAY, @DATE)),@DATE))),0)



1. 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