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

Если строки будут с абсолютно разными значениями в столбце, к которому применяется ранжирующая оконная функция.

1. Решите на базе данных AdventureWorks2017 следующие задачи.

1. Изучите данные в таблице Production.UnitMeasure. Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’. Сколько всего различных кодов здесь есть?

Вставьте следующий набор данных в таблицу:

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

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

SELECT \*

FROM [Production].[UnitMeasure]

SELECT [UnitMeasureCode]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'

SELECT DISTINCT COUNT ([UnitMeasureCode])

FROM [Production].[UnitMeasure]

INSERT INTO [Production].[UnitMeasure]

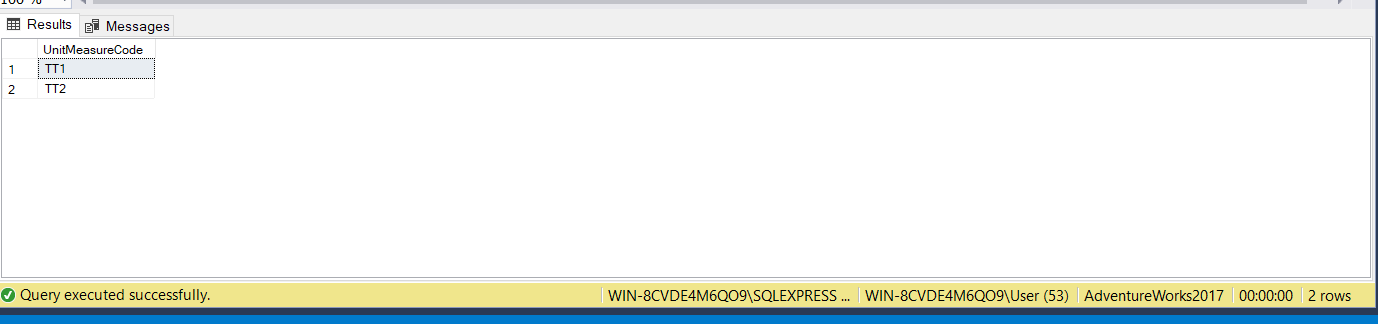
VALUES ('TT1', 'Test 1', CONVERT ( datetime, '09-09-2020', 120)),

('TT2', 'Test 2', GETDATE () )

SELECT [UnitMeasureCode]

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'



1. Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’.  Посмотрите результат в отсортированном виде по коду.

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

INTO Production.UnitMeasureTest

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%'

SELECT \* FROM Production.UnitMeasureTest

INSERT INTO Production.UnitMeasureTest

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

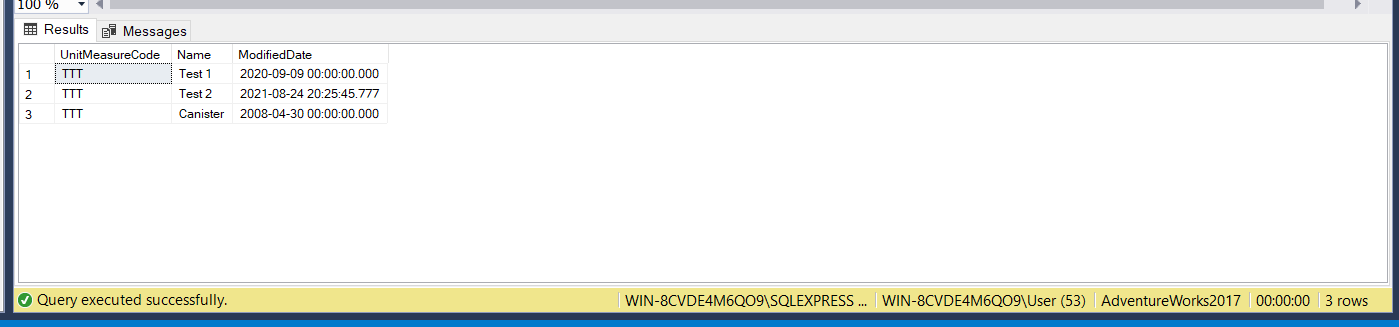
FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] = 'CAN'

SELECT \*

FROM Production.UnitMeasureTest

ORDER BY [UnitMeasureCode]

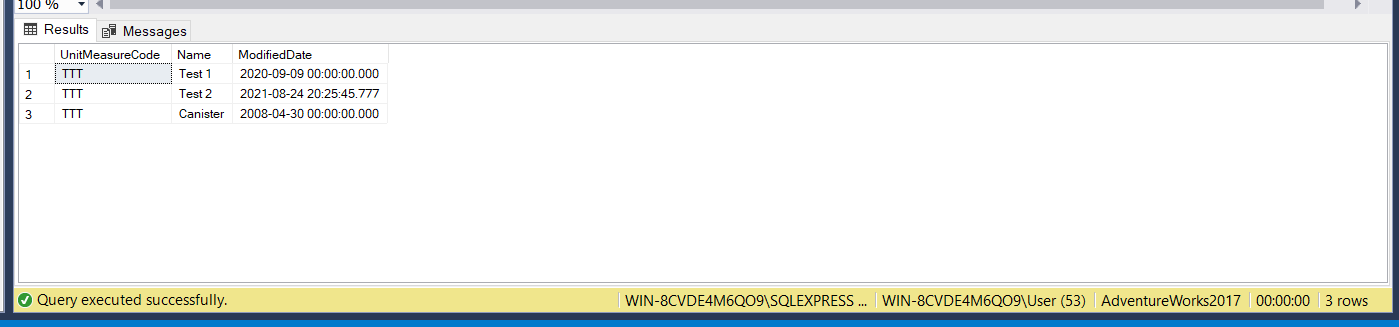


Тут Я поздно сообразил, что скринов надо было наделать. Уже обновил данные в таблице для следующего задания, поэтому результат запроса выглядит не так как должен быть.

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

UPDATE Production.UnitMeasureTest

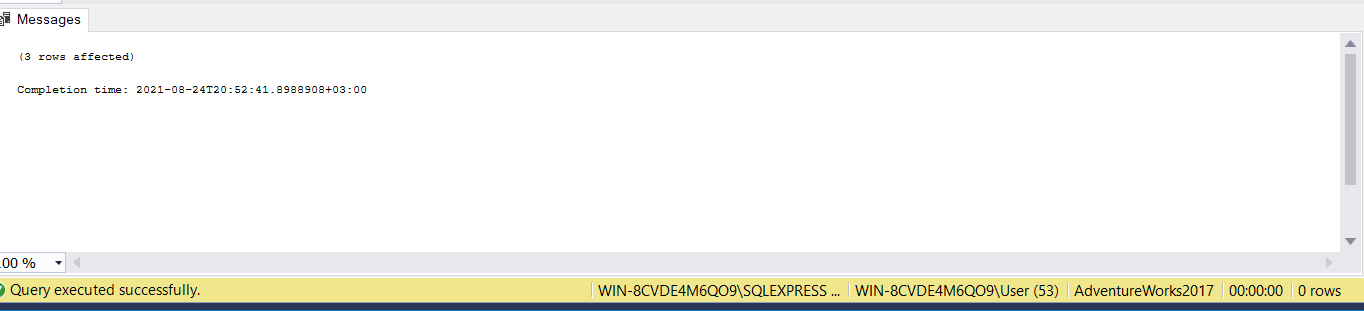
SET [UnitMeasureCode] = 'TTT'



d. Удалите все строки из Production.UnitMeasureTest.

DELETE

FROM Production.UnitMeasureTest



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

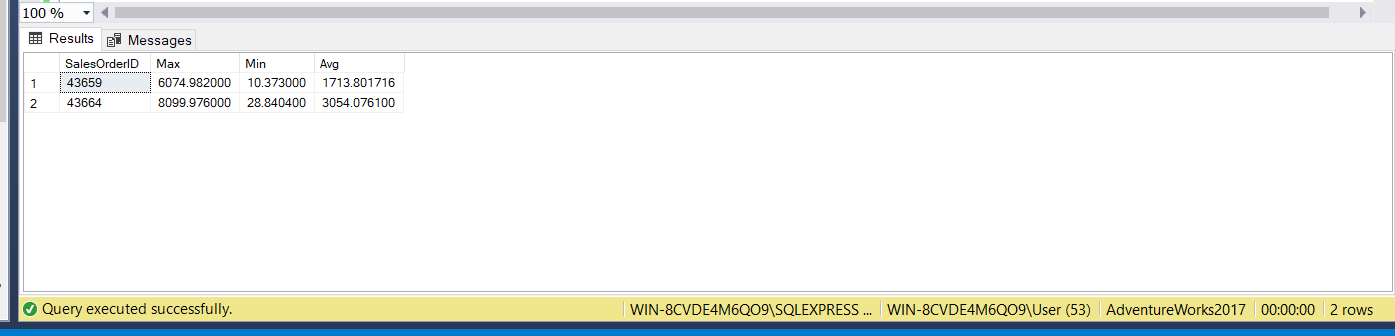
SELECT DISTINCT [SalesOrderID],

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]

WHERE [SalesOrderID] IN (43659, 43664)

f. 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).

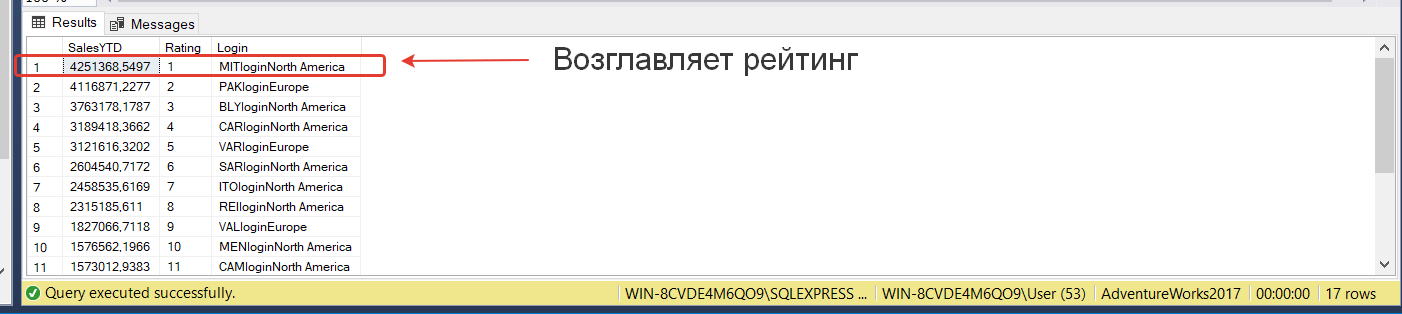
Кто возглавляет рейтинг? А кто возглавлял рейтинг в прошлом году (SalesLastYear).

SELECT SalesYTD,

DENSE\_RANK() OVER (ORDER BY SalesYTD DESC) AS "Rating",

CONCAT (UPPER(LEFT(LastName,3)),'login',TerritoryGroup ) AS "Login"

FROM Sales.vSalesPerson

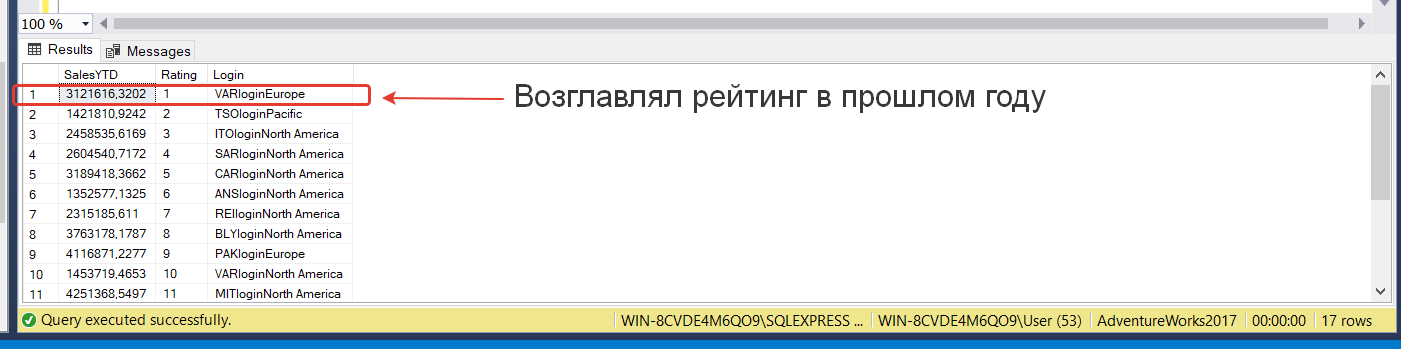


SELECT SalesYTD,

DENSE\_RANK() OVER (ORDER BY SalesLastYear DESC) AS "Rating",

CONCAT (UPPER(LEFT(LastName,3)),'login',TerritoryGroup ) AS "Login"

FROM Sales.vSalesPerson



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

SELECT

CASE

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

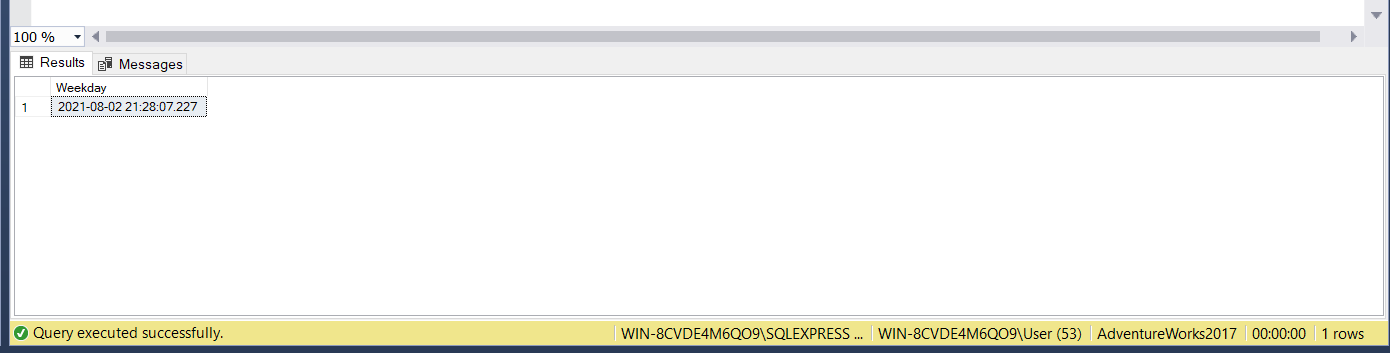
THEN DATEADD(dd, -DATEPART(dd, GETDATE()) + 3, GETDATE())

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

THEN DATEADD(dd, -DATEPART(dd, GETDATE()) + 2, GETDATE())

ELSE DATEADD(dd, -DATEPART(dd, GETDATE()) + 1, GETDATE())

END 'Weekday'



1. 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