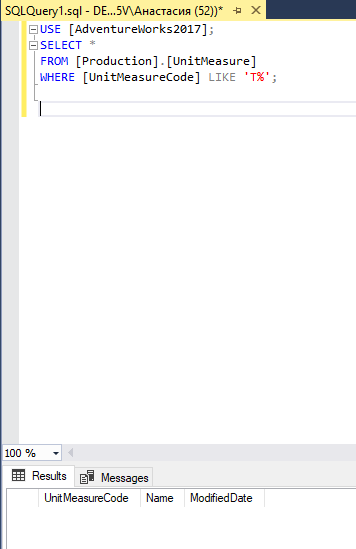
1. Решите на базе данных AdventureWorks2017 следующие задачи.
2. Изучите данные в таблице Production.UnitMeasure. Проверьте, есть ли здесь UnitMeasureCode, начинающиеся на букву ‘Т’.

USE [AdventureWorks2017];

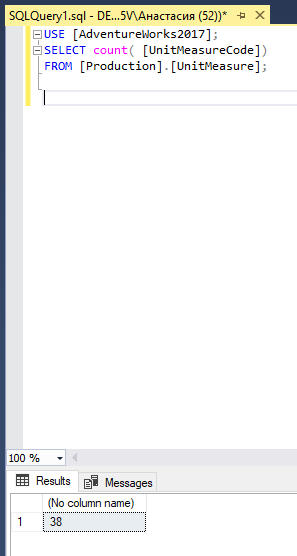
SELECT \*

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 'T%';



Сколько всего различных кодов здесь есть?



Вставьте следующий набор данных в таблицу:

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

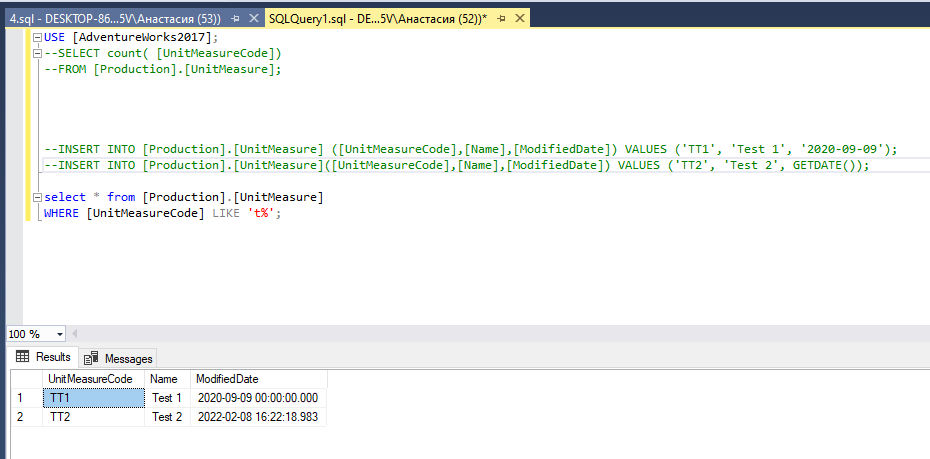
INSERT INTO [Production].[UnitMeasure] ([UnitMeasureCode],[Name],[ModifiedDate]) VALUES ('TT1', 'Test 1', '2020-09-09');

INSERT INTO [Production].[UnitMeasure]([UnitMeasureCode],[Name],[ModifiedDate]) VALUES ('TT2', 'Test 2', GETDATE());

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

SELECT \* FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 't%';

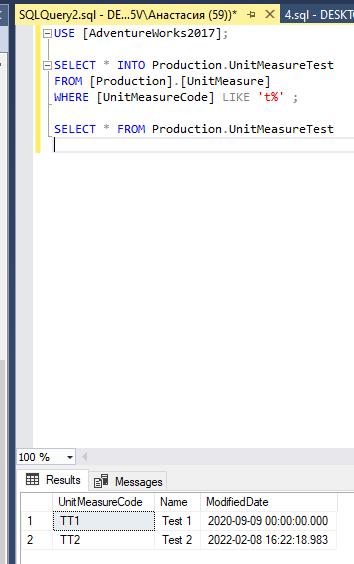


1. Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. в отсортированном виде по коду.

SELECT \* INTO Production.UnitMeasureTest

FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] LIKE 't%';

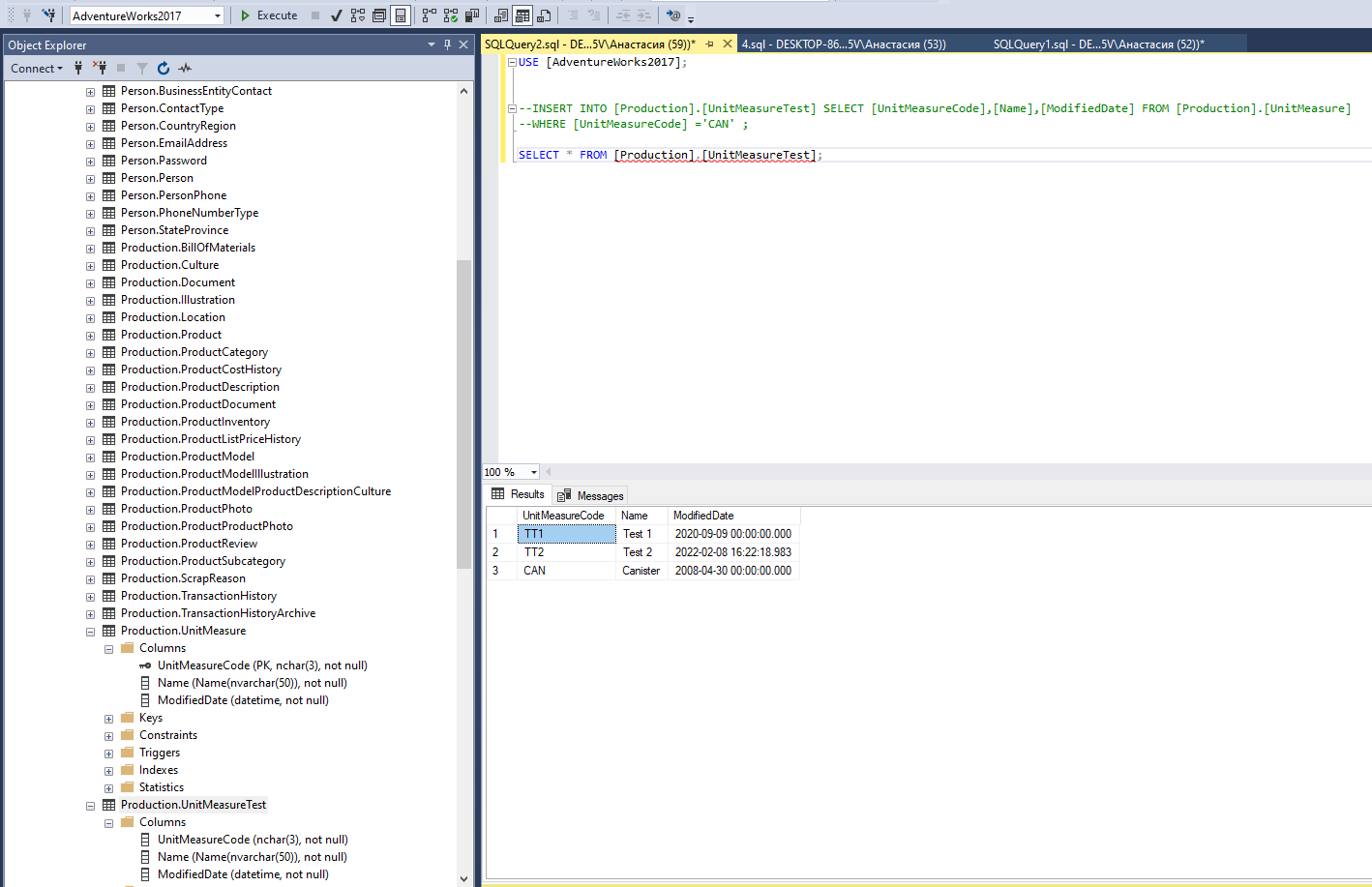


Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’. Посмотрите результат.

INSERT INTO [Production].[UnitMeasureTest] SELECT [UnitMeasureCode],[Name],[ModifiedDate] FROM [Production].[UnitMeasure]

WHERE [UnitMeasureCode] ='CAN' ;

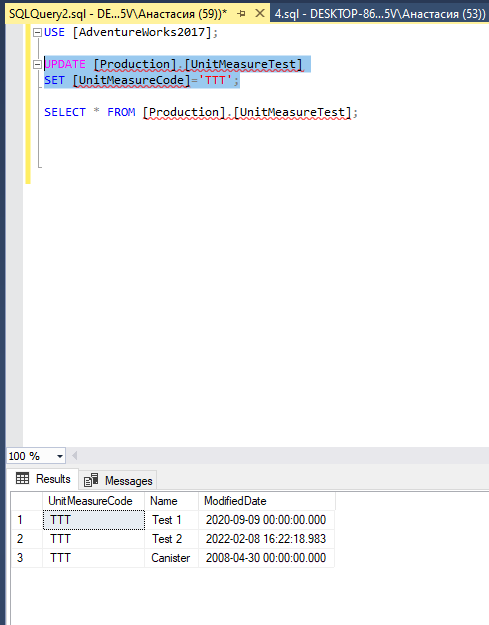
SELECT \* FROM [Production].[UnitMeasureTest];



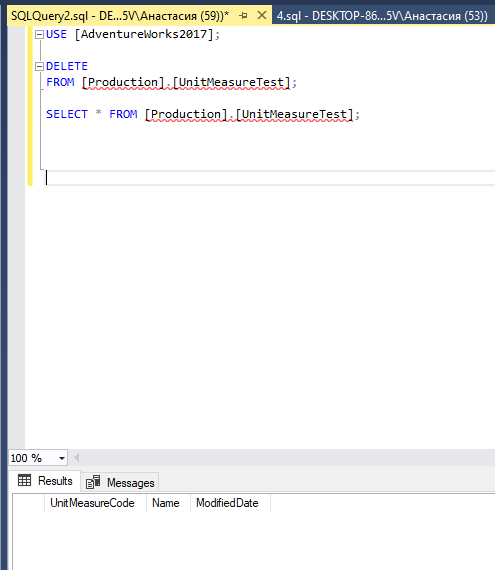
1. Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’.

UPDATE [Production].[UnitMeasureTest]

SET [UnitMeasureCode]='TTT';



1. Удалите все строки из 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, ProductID, LineTotal

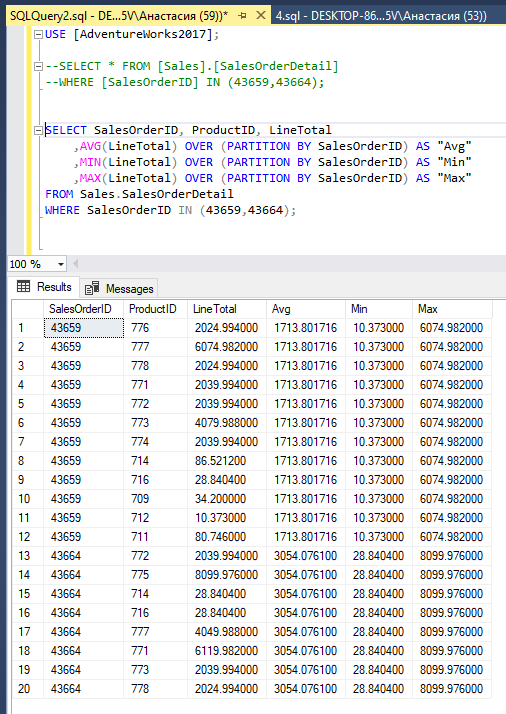
,AVG(LineTotal) OVER (PARTITION BY SalesOrderID) AS "Avg"

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

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

FROM Sales.SalesOrderDetail

WHERE SalesOrderID IN (43659,43664);



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).

Кто возглавляет рейтинг? 276

select \* from (

SELECT pp.BusinessEntityID,pp.LastName,st.[Group],concat(upper(left(pp.LastName,3)),'login',isnull(st.[Group],'')) as login,sp.SalesYTD,

RANK() over (order by sp.SalesYTD desc) as rating

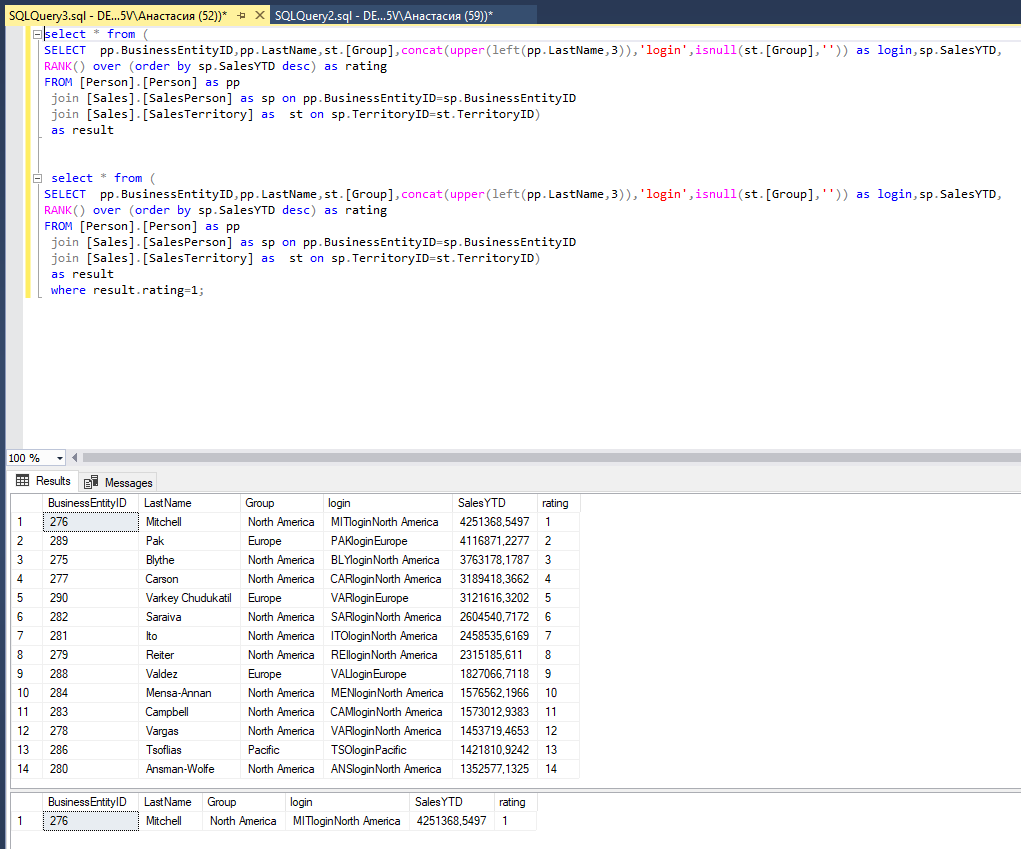
FROM [Person].[Person] as pp

join [Sales].[SalesPerson] as sp on pp.BusinessEntityID=sp.BusinessEntityID

join [Sales].[SalesTerritory] as st on sp.TerritoryID=st.TerritoryID)

as result

where result.rating=1;



А кто возглавлял рейтинг в прошлом году (SalesLastYear). 290

select \* from (

SELECT pp.BusinessEntityID,pp.LastName,st.[Group],concat(upper(left(pp.LastName,3)),'login',isnull(st.[Group],'')) as login,sp.SalesYTD,

RANK() over (order by sp.SalesLastYear desc) as rating

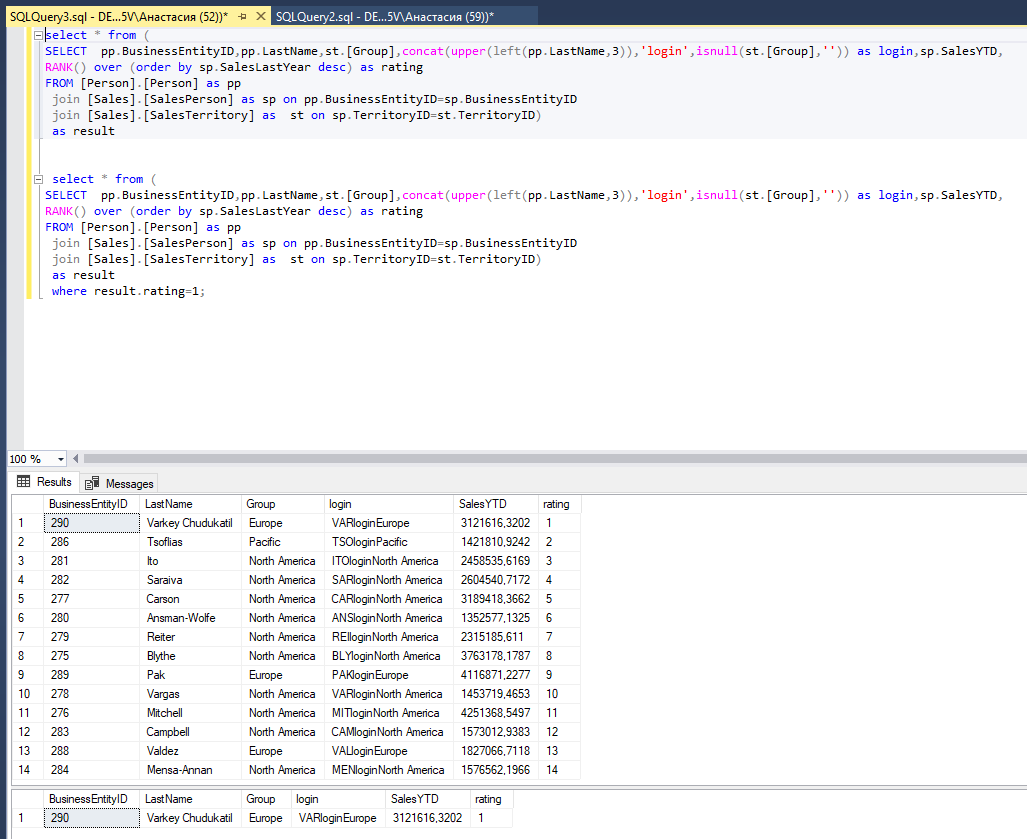
FROM [Person].[Person] as pp

join [Sales].[SalesPerson] as sp on pp.BusinessEntityID=sp.BusinessEntityID

join [Sales].[SalesTerritory] as st on sp.TerritoryID=st.TerritoryID)

as result

where result.rating=1;



1. Найдите первый будний день месяца (FROM не используем). Нужен

стандартный код на все времена.

Я двигалась в правильном направлении, но до конца так и не дошла.

set datefirst 1;

select DATEADD(DAY,1,EOMONTH(GETDATE(),-1)) AS FirstDayCurMonth,

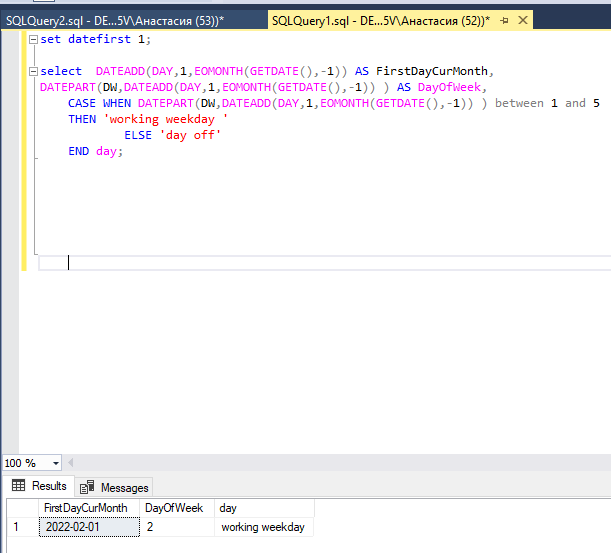
DATEPART(DW,DATEADD(DAY,1,EOMONTH(GETDATE(),-1)) ) AS DayOfWeek,

CASE WHEN DATEPART(DW,DATEADD(DAY,1,EOMONTH(GETDATE(),-1)) ) between 1 and 5

THEN 'working weekday '

ELSE 'day off'

END day;



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

Применяя Rank правило такое - если несколько строк имеют одинаковый ранг, то следующий ранг будет с разрывом в значении.

DENSE\_RANK генерирует уникальный идентификатор для каждой группы без разрывов.

Получается, что при неповторяющихся(разных) значениях (как в примере ниже с col\_value )

оконные функции Row Number, Rank и Dense Rank вернут одинаковый результат.

SELECT

Col\_Value,

ROW\_NUMBER() OVER (ORDER BY Col\_Value) AS 'RowID',

Rank() OVER (ORDER BY Col\_Value) AS 'Rank',

DENSE\_RANK() OVER (ORDER BY Col\_Value) AS 'dense\_rank'

FROM

Table col;

col\_value RowId Rank dense\_rank

A 1 1 1

B 2 2 2

C 3 3 3

D 4 4 4

E 5 5 5

F 6 6 6

G 7 7 7

А также по идее одинаковый результат будет в том случае, если в таблице всего 1 строка.

5. 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

count (поле)- подсчитает количество записей, поле которых не равно нулю.