**Домашняя работа 4**

Углубленный уровень SQL

**Задания**

3/ Посмотрите, какие ещё оконные функции cуществуют:

* [Ранжирующие функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/ranking-functions-transact-sql?view=sql-server-ver15)
* [Агрегатные функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/aggregate-functions-transact-sql?view=sql-server-ver15)
* [Аналитические функции](https://docs.microsoft.com/ru-ru/sql/t-sql/functions/analytic-functions-transact-sql?view=sql-server-ver15)

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

- при отсутствии повторяющихся значений в строках

4/ 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’.

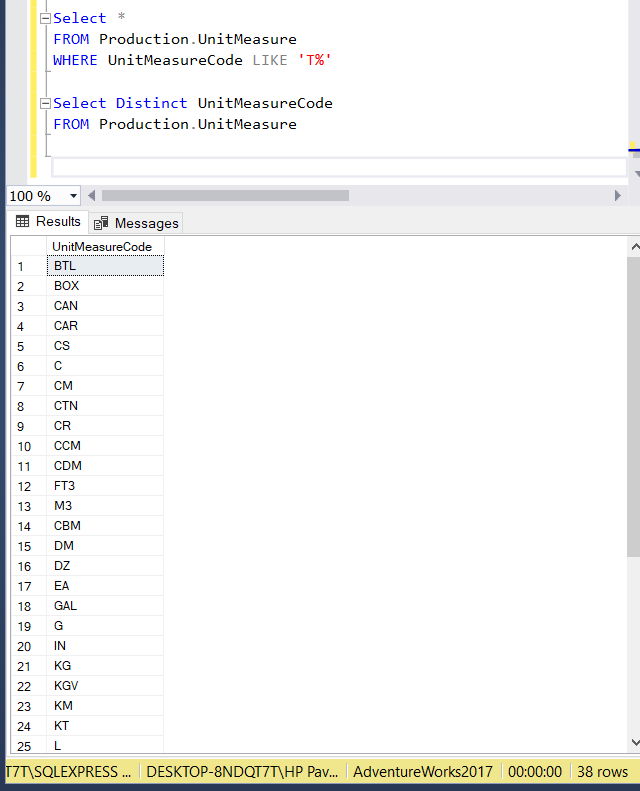
Select \*

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'

Select Distinct UnitMeasureCode

FROM Production.UnitMeasure

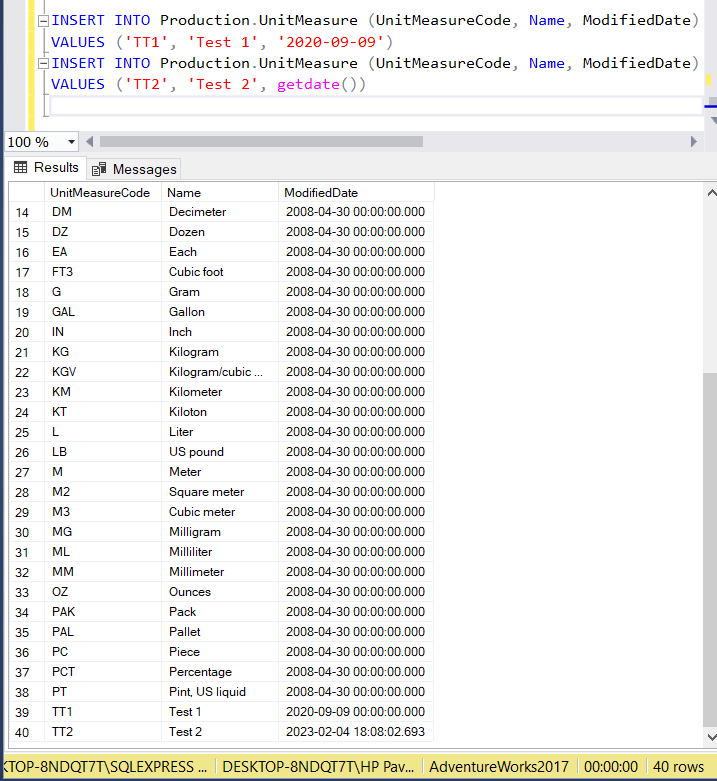


INSERT INTO Production.UnitMeasure (UnitMeasureCode, Name, ModifiedDate)

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

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

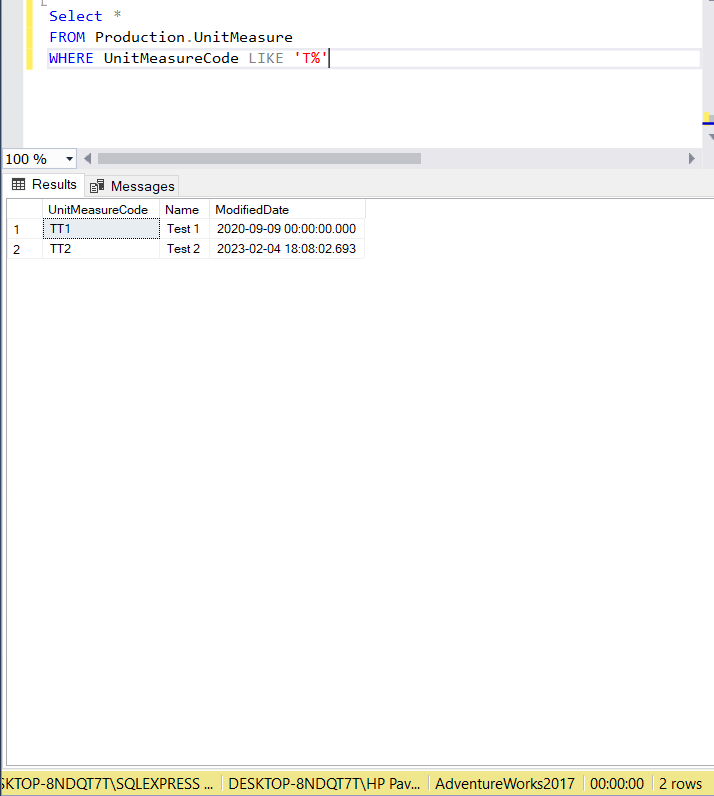
VALUES ('TT2', 'Test 2', getdate())



Select \*

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'



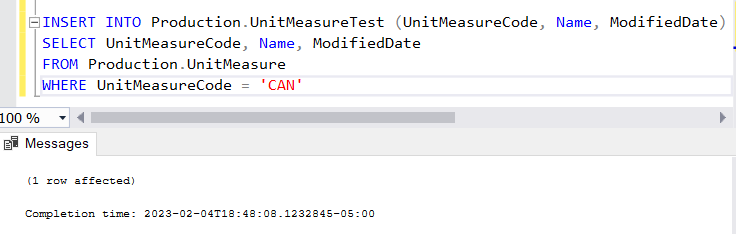
1. Теперь загрузите вставленный набор в новую, не существующую таблицу Production.UnitMeasureTest. Догрузите сюда информацию из Production.UnitMeasure по UnitMeasureCode = ‘CAN’. Посмотрите результат в отсортированном виде по коду.

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

SELECT UnitMeasureCode, Name, ModifiedDate

FROM Production.UnitMeasure

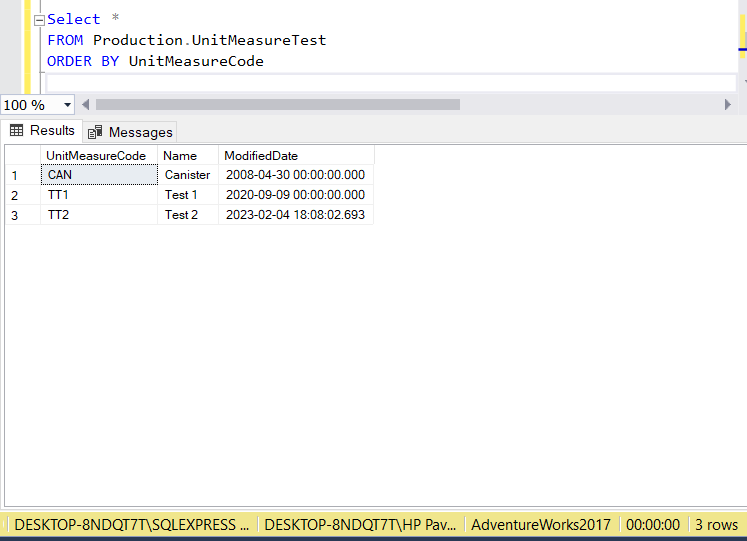
WHERE UnitMeasureCode = 'CAN'



Select \*

FROM Production.UnitMeasureTest

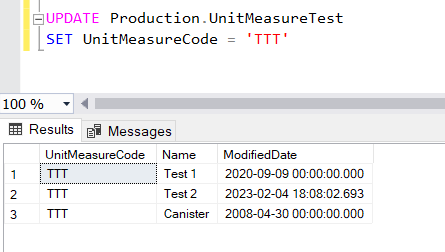
ORDER BY UnitMeasureCode



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

UPDATE Production.UnitMeasureTest

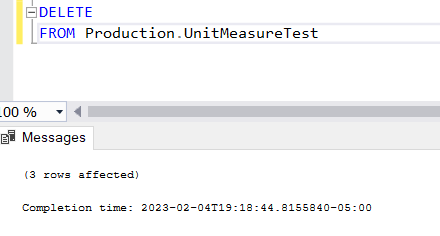
SET UnitMeasureCode = 'TTT'



1. Удалите все строки из Production.UnitMeasureTest.

DELETE

FROM Production.UnitMeasureTest



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

SELECT SalesOrderID, LineTotal,

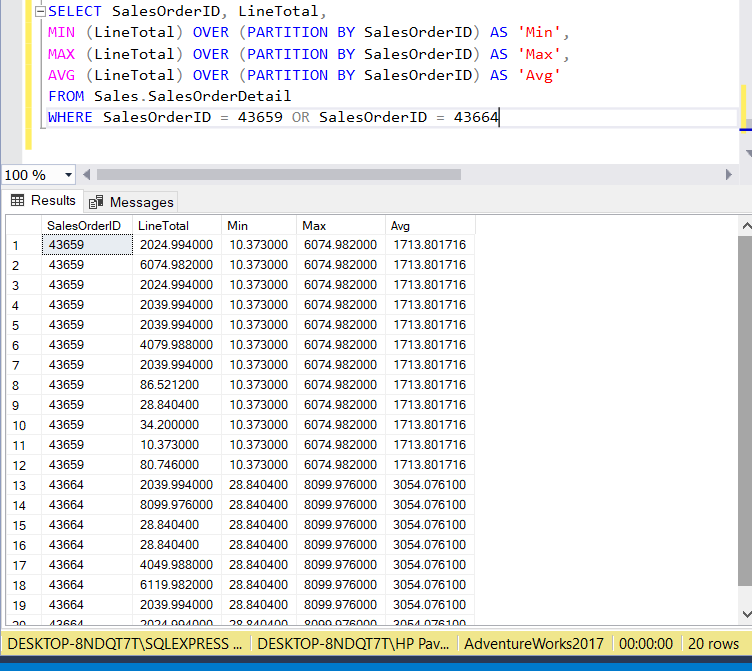
MIN (LineTotal) OVER (PARTITION BY SalesOrderID) AS 'Min',

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

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

FROM Sales.SalesOrderDetail

WHERE SalesOrderID = 43659 OR SalesOrderID = 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? А кто возглавлял рейтинг в прошлом году (SalesLastYear).

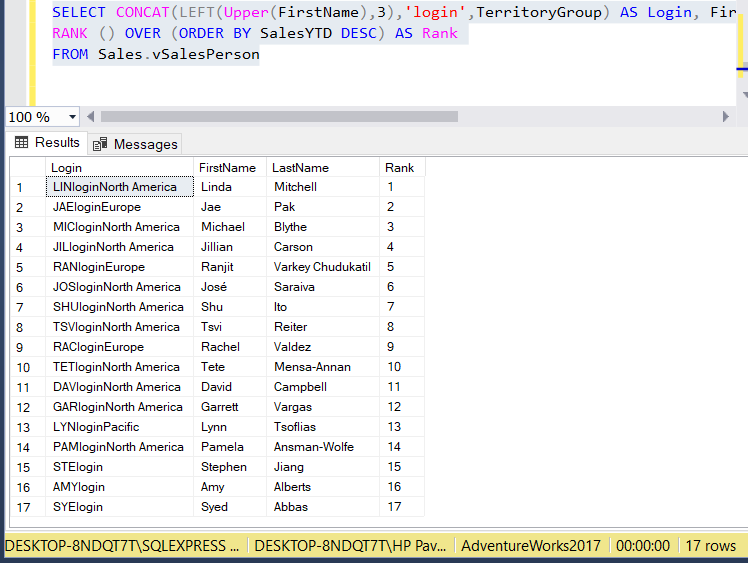
* Возглавляет рейтинг Linda Mitchel
* В прошлом году Ranjit Varkey Chudukatil

SELECT CONCAT(LEFT(Upper(FirstName),3),'login',TerritoryGroup) AS Login,

FirstName, LastName,

RANK () OVER (ORDER BY SalesYTD DESC) AS Rank

FROM Sales.vSalesPerson

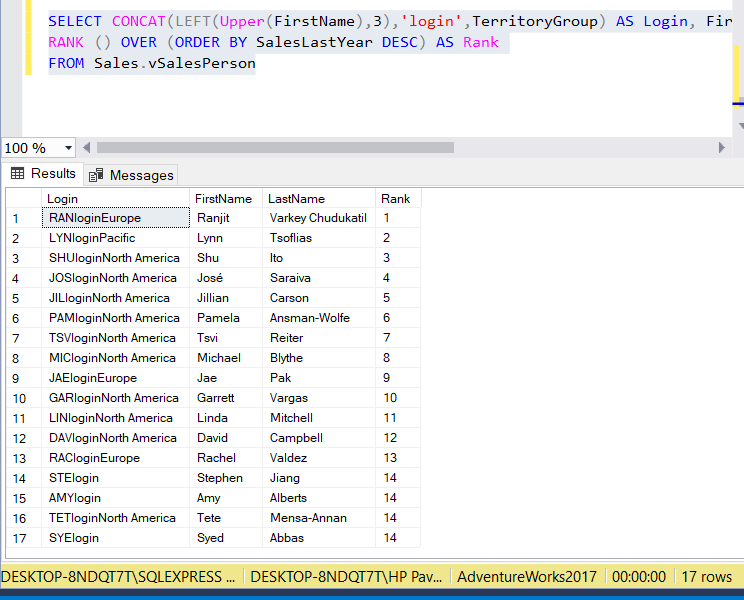


SELECT CONCAT(LEFT(Upper(FirstName),3),'login',TerritoryGroup) AS Login,

FirstName, LastName,

RANK () OVER (ORDER BY SalesLastYear DESC) AS Rank

FROM Sales.vSalesPerson



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

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