## **--4A**

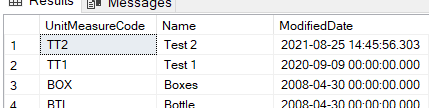
### ***--Изучите данные в таблице Production.UnitMeasure.***

SELECT \*

FROM Production.UnitMeasure

ORDER BY ModifiedDate desc

**Result:**

****

### --4A/1

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

**Code:**

SELECT \*

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'

**Result:**



### --4A/2

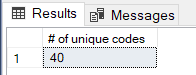
### --Сколько всего различных кодов здесь есть?

### Code:

SELECT COUNT(DISTINCT UnitMeasureCode)

FROM Production.UnitMeasure

**Result:**



### --4A/3

### --Вставьте следующий набор данных в таблицу:

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

**Code:**

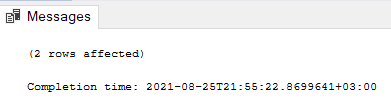
INSERT INTO AdventureWorks2017.Production.UnitMeasure

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

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

UnitMeasureCode Name ModifiedDate

**Result:**



## --4B

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

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

**Code:**

SELECT UnitMeasureCode, Name, ModifiedDate

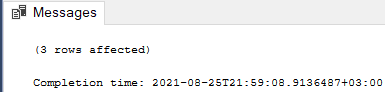
INTO Production.UnitMeasureTest

FROM Production.UnitMeasure

WHERE UnitMeasureCode LIKE 'T%'

OR UnitMeasureCode = 'CAN'

**Result:**



### --Посмотрите результат в отсортированном виде по коду.

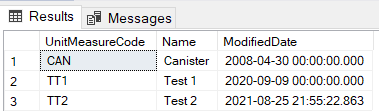
**Code:**

SELECT UnitMeasureCode, Name, ModifiedDate

FROM Production.UnitMeasureTest

ORDER BY UnitMeasureCode

**Result:**



## --4C

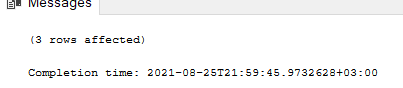
### --Измените UnitMeasureCode для всего набора из Production.UnitMeasureTest на ‘TTT’.

**Code:**

UPDATE Production.UnitMeasureTest

SET UnitMeasureCode = 'TTT'

**Result:**



## --4D

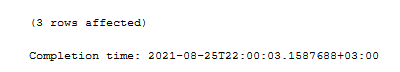
### --Удалите все строки из Production.UnitMeasureTest.

**Code:**

DELETE

FROM Production.UnitMeasureTest

**Result:**



## --4E

### --Найдите информацию из Sales.SalesOrderDetail по заказам 43659,43664.

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

**Comment:** I wasn’t sure if we had to add max, min, avg to all other column or not, but it doesn’t matter much I suppose, otherwise would just need to separate it into 2 different queries.

**Code:**

SELECT \*, MAX(LineTotal) OVER (PARTITION BY SalesOrderID) as MaxLineTotal,

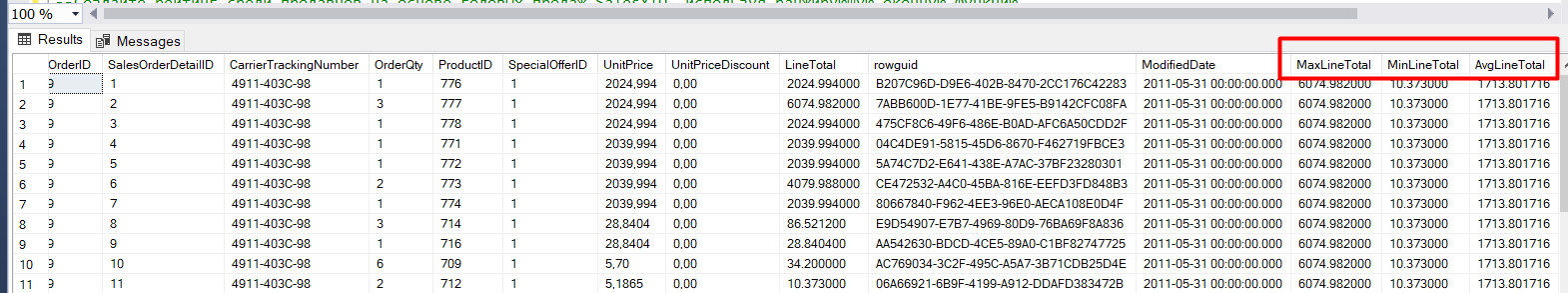
MIN(LineTotal) OVER (PARTITION BY SalesOrderID) as MinLineTotal,

AVG(LineTotal) OVER (PARTITION BY SalesOrderID) as AvgLineTotal

FROM Sales.SalesOrderDetail

WHERE SalesOrderID = 43659 OR SalesOrderID = 43664

**Result:**



## --4F

### --Изучите данные в объекте Sales.vSalesPerson.

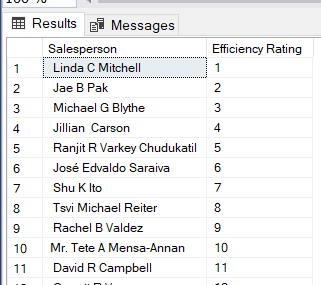
### --Создайте рейтинг cреди продавцов на основе годовых продаж SalesYTD, используя ранжирующую оконную функцию.

**Code:**

SELECT CONCAT (Title, ' ', FirstName, ' ', MiddleName, ' ', LastName), DENSE\_RANK() OVER (ORDER BY SalesYTD desc) AS "Efficiency Rating"

FROM Sales.vSalesPerson

**Result:**



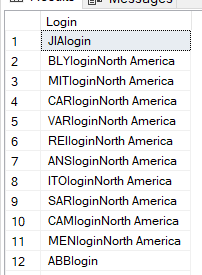
### --Добавьте поле Login, состоящий из 3 первых букв фамилии в верхнем регистре + ‘login’ + TerritoryGroup (Null заменить на пустое значение).

**Code:**

SELECT CONCAT (UPPER(SUBSTRING(LastName, 1, 3)), 'login', ISNULL(TerritoryGroup,' ')) AS 'Login'

FROM Sales.vSalesPerson

**Result:**



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

**Code:**

SELECT CONCAT (Title, ' ', FirstName, ' ', MiddleName, ' ', LastName),

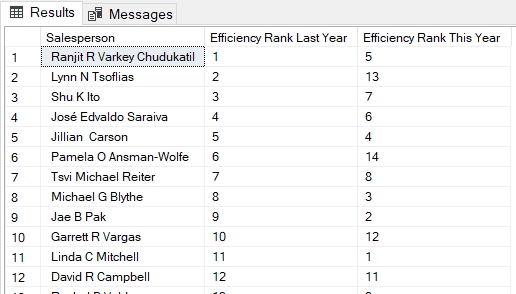
DENSE\_RANK() OVER (ORDER BY SalesLastYear desc) AS "Efficiency Rating Last Year",

DENSE\_RANK() OVER (ORDER BY SalesYTD desc) AS "Efficiency Rating This Year"

FROM Sales.vSalesPerson

ORDER BY "Efficiency Rating Last Year" --/"Efficiency Rating This Year" asc

**Result:**



## --4G

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

**Code:**

SELECT

CASE

WHEN DATENAME(weekday, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1)) = 'Saturday' THEN DATEADD(WEEKDAY, 2, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1))

WHEN DATENAME(weekday, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1)) = 'Sunday' THEN DATEADD(WEEKDAY, 1, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1))

ELSE DATEADD(WEEKDAY, 1, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1))

END AS 'First Working Day In The Month',

CASE

WHEN DATENAME(weekday, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1)) = 'Saturday' THEN 'Saturday'

WHEN DATENAME(weekday, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1)) = 'Sunday' THEN 'Sunday'

ELSE DATENAME(weekday, DATEFROMPARTS(YEAR(GETDATE()),MONTH(GETDATE()),1))

END AS 'Day of the week'

**Comment:** I’ve decided to also add a column ‘Day of the week’, so that we are sure the query works correctly and no weekend days are shown as working days. Also I suppose we could filter the date based on the calendar of country’s holidays or something, but I think it’s not for this task.

**Result:**



## --5

--Посчитать count(1), count(name), count(id), count(\*)

Id(PK) Name DepName

1 null A

2 null null

3 A C

4 B C

**Result:**

SELECT count(1), count(name), count(id), count(\*)

Count1 Count(Name) Count(id) Count(\*)

4 2 4 4