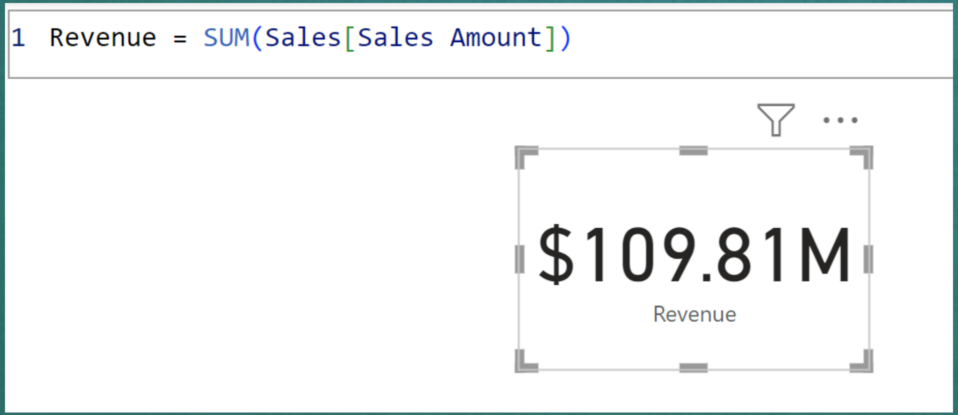
SUM

- It adds all the values in a column and the column should be numeric type.



```
Revenue = SUM(Sales[Sales Amount])
```

Karthimeena

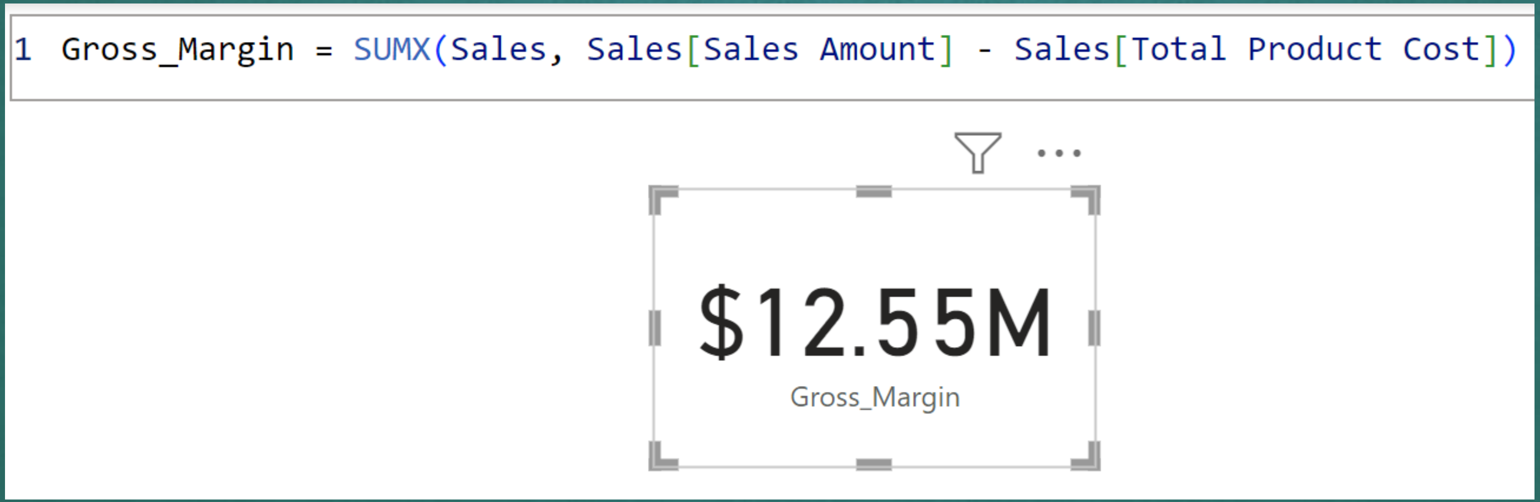
SUMX

- This useful to performing calculations based on applied expressions on a row-by-row basis.
- It accepts table as first argument and second argument is a column that contains a value you want to sum.
- It returns a decimal number.
- This function ignores if there is BLANK and logical values while adding numbers.

Karthimeena

SUMX

- This function returns the gross margin amount by calculating an expression on a row-by-row basis.

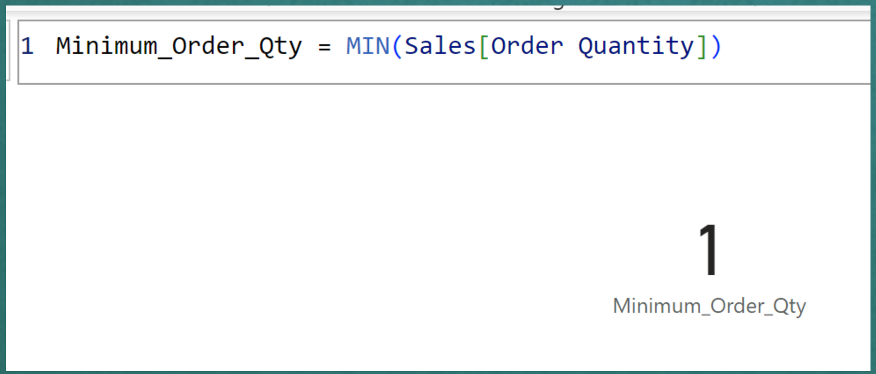


```
Gross_Margin = SUMX(Sales, Sales[Sales Amount] - Sales[Total Product Cost])
```

Karthimeena

MIN

- This function is helpful for finding the minimum value of a given column.



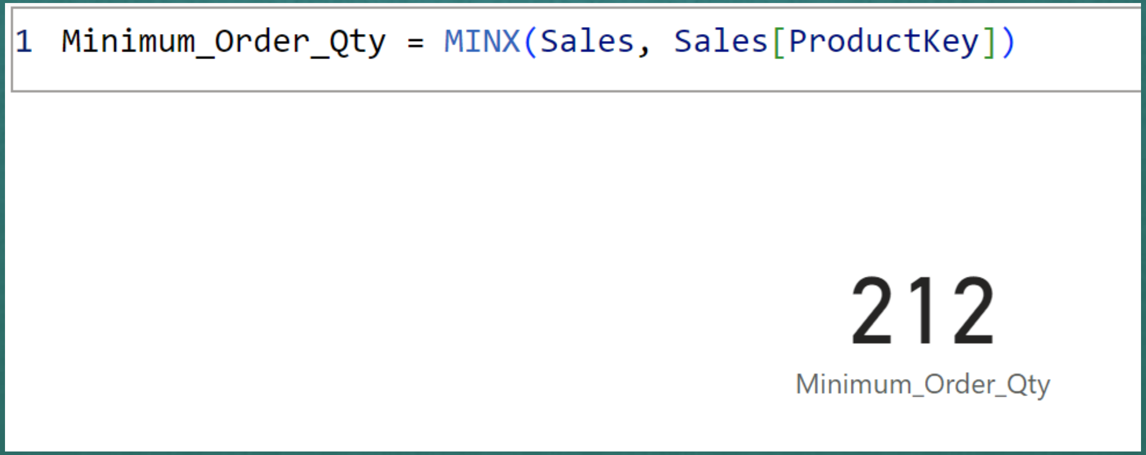
```
Minimum_Order_Qty = MIN(Sales[Order Quantity])
```

```
1
Minimum_Order_Qty
```

Karthimeena

MINX

- This calculates the minimum value based on some filters based on some expression for each row in table.
- It accepts table as first argument, expression or filters as second argument.



```
Minimum_Order_Qty = MINX(Sales, Sales[ProductKey])
```

212

Minimum_Order_Qty

Karthimeena

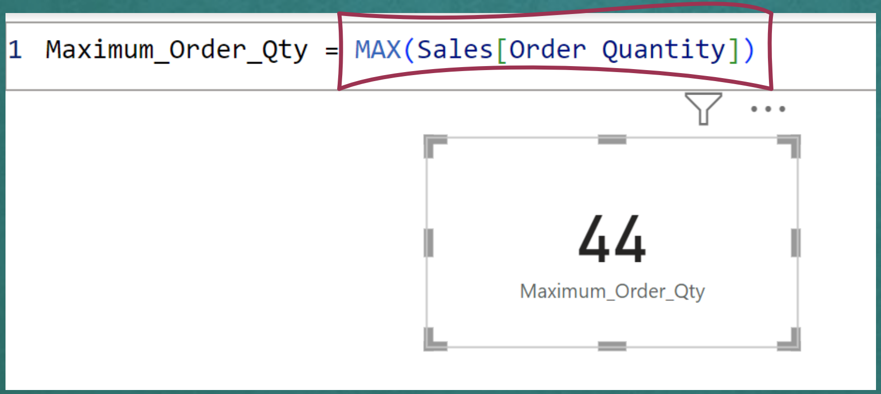
MINA

- This function performs as MIN with added feature to handle the non-numeric columns
- This calculates the minimum value of the column including number, text and date.
- Minimum_Order_Qty = MINA('Sales Order'[Sales Order Line])
- This returns value 0, since the data type of the column is TEXT, it does not find any value

Karthimeena

MAX

- This function is helpful for finding the greatest value of a given column.



```
Maximum_Order_Qty = MAX(Sales[Order Quantity])
```

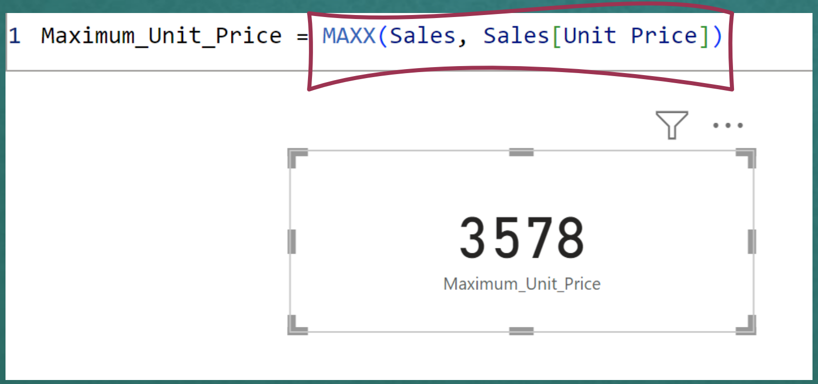
44

Maximum_Order_Qty

Karthimeena

MAXX

- This function calculates the maximum values of a given column based on expression for each row in a table.
- It accepts table as first argument, expression or filters as second argument.
- This evaluates number, date and Text.



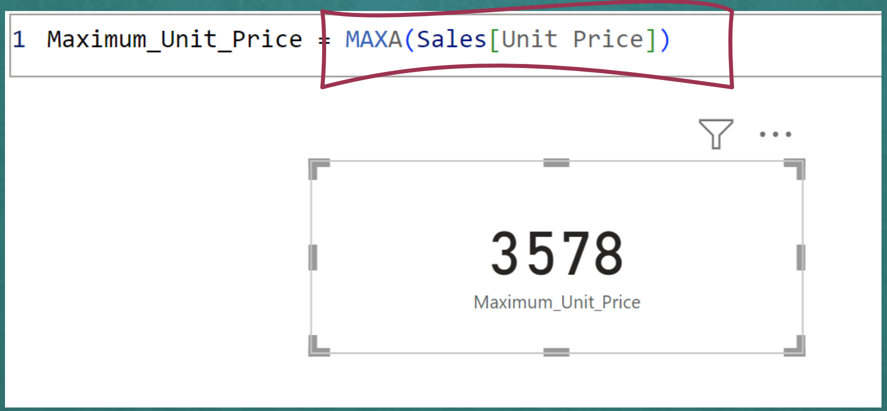
```
Maximum_Unit_Price = MAXX(Sales, Sales[Unit Price])
```

3578
Maximum_Unit_Price

Karthimeena

MAXA

- This function finds largest values of the column.
- It handles the non-numeric columns such as number and date.
- This calculates the maximum value of the column including number, text and date.



```
Maximum_Unit_Price = MAXA(Sales[Unit Price])
```

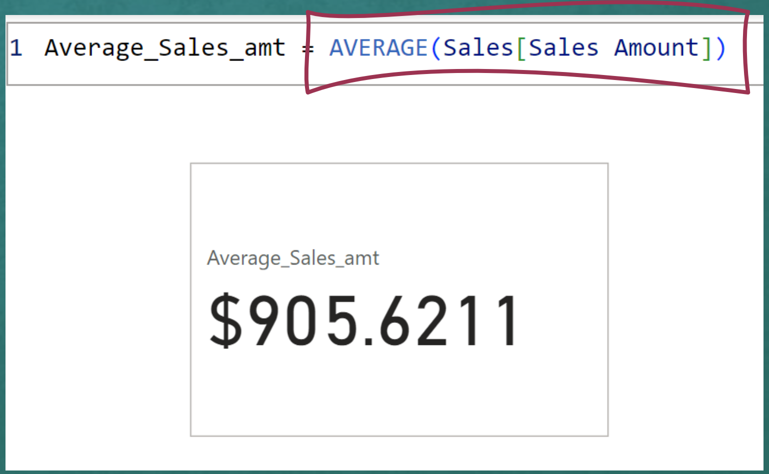
3578
Maximum_Unit_Price

- MAXA() returns value 0, if the data type of the column is TEXT, it does not find any value

Karthimeena

AVERAGE

- This function calculates the arithmetic average operation of all the numbers in a column and returns the result.
- If the column contains value 0 are included.
- If the column contains TEXT, this won't perform operation and just returns the blank value.



```
Average_Sales_amt = AVERAGE(Sales[Sales Amount])
```

Average_Sales_amt

$905.6211

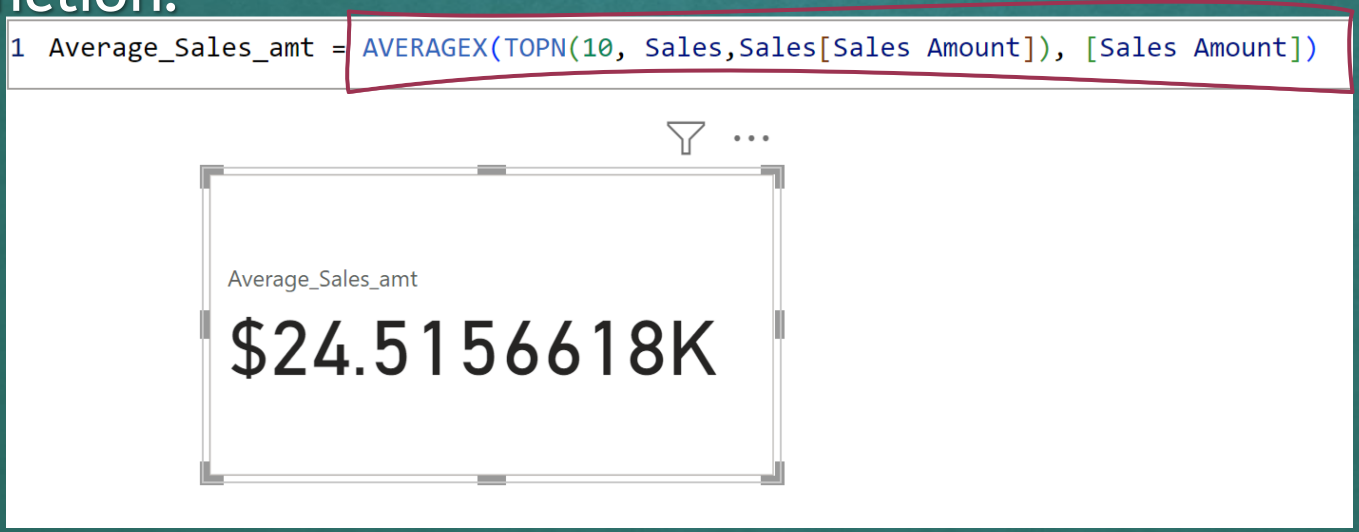
Karthimeena

AVERAGEX

- If you want to perform average operation for a set of numbers instead of entire column in table, this AVERAGEX will do the same.
- Function takes table as first argument and expression as second argument.
- You cannot include non-numeric values and both arguments are required.

AVERAGEX

- This DAX code calculates the average value of top 10 numbers in the table.
- TopN function returns the top N values as a table to AVERAGEX function.

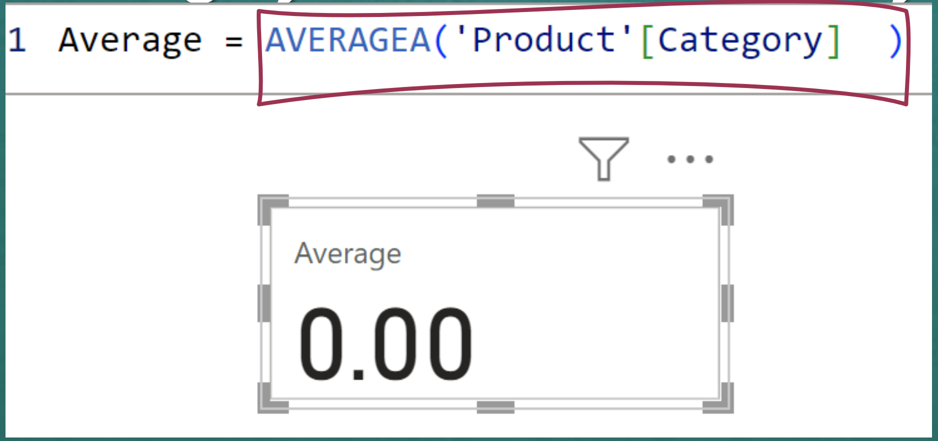


```
Average_Sales_amt = AVERAGEX(TOPN(10, Sales,Sales[Sales Amount]), [Sales Amount])
```

Karthimeena

AVERAGEA

- This returns the average of values in a column and it handled the non-numeric columns.
- Values that contain non-numeric text,0, empty (“”) are treated as 0 when calculating the average.
- The column ‘Category’ does not contain any numeric values.



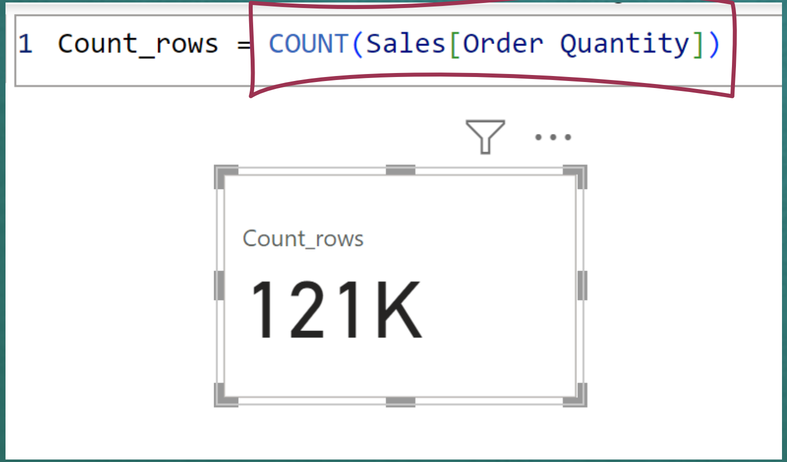
```
Average = AVERAGEA('Product'[Category]  )
```

Average
0.00

Karthimeena

COUNT

- This returns number of rows in a given column that contains non-blank values.
- It counts the string, date, number.
- COUNT() accepts a column as an arguments and returns integer value.



```
Count_rows = COUNT(Sales[Order Quantity])
```

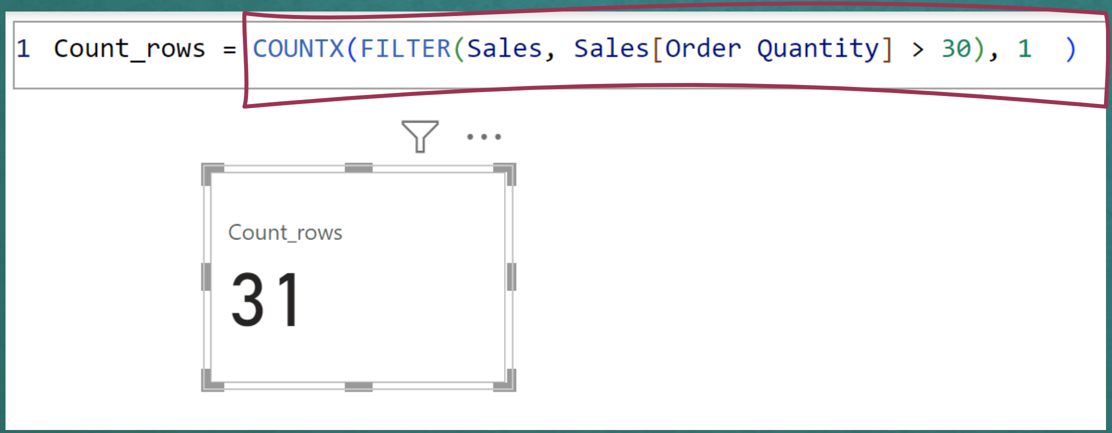
Karthimeena

COUNTX

- This used to count the number of rows in a table or table expression that satisfies a specified condition.
- COUNTX() accepts table as an first arguments and expression as an second argument.
- It count only non-blank values from the expression.

COUNTX

- Below code, uses filter as the first argument and ‘1’ as expression to count each rows from filtered table.



```
Count_rows = COUNTX(FILTER(Sales, Sales[Order Quantity] > 30), 1  )
```

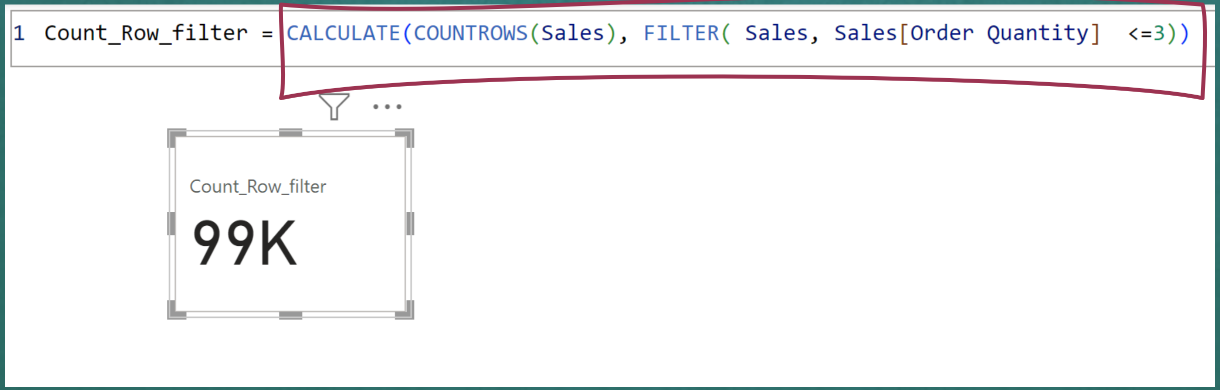
Count_rows

31

Karthimeena

COUNTROWS

- COUNTROWS() used to count the number of rows in the table or given expression.
- The parameter is optional here.
- If parameter is not given, it takes default value as the home table of current context

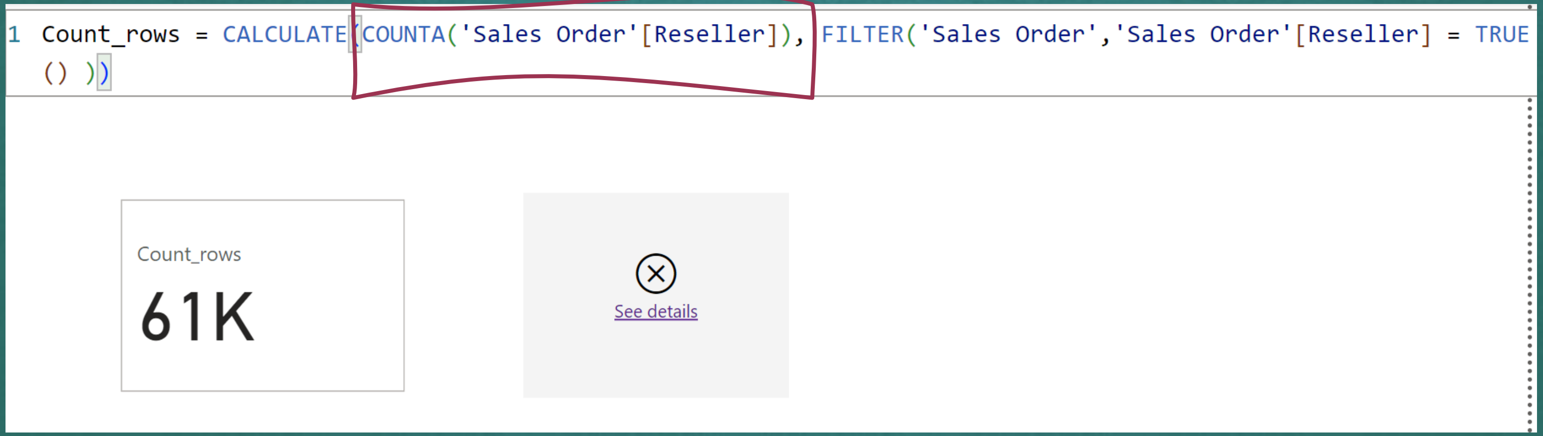


```
Count_Row_filter = CALCULATE(COUNTROWS(Sales), FILTER( Sales, Sales[Order Quantity]  <=3))
```

Karthimeena

COUNTA

- COUNTA() used to count the number of rows in the table contains non-blank values.
- COUNTA() supports BOOLEAN data type.

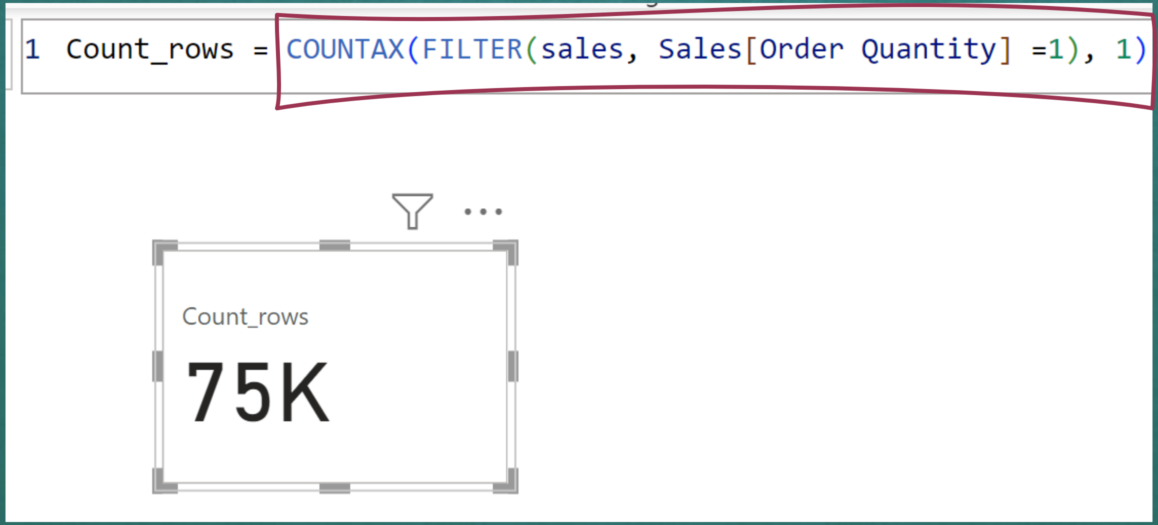


```
Count_rows = CALCULATE(COUNTA('Sales Order'[Reseller]), FILTER('Sales Order','Sales Order'[Reseller] = TRUE
() ))
```

Karthimeena

COUNTAX

- COUNTAX() function counts the number of rows of a given expression in the context.
- The following code counts the number of rows from filtered table where order quantity is equal to 1 and uses 1 as a expression to count the rows.



```
Count_rows = COUNTAX(FILTER(sales, Sales[Order Quantity] =1), 1)
```

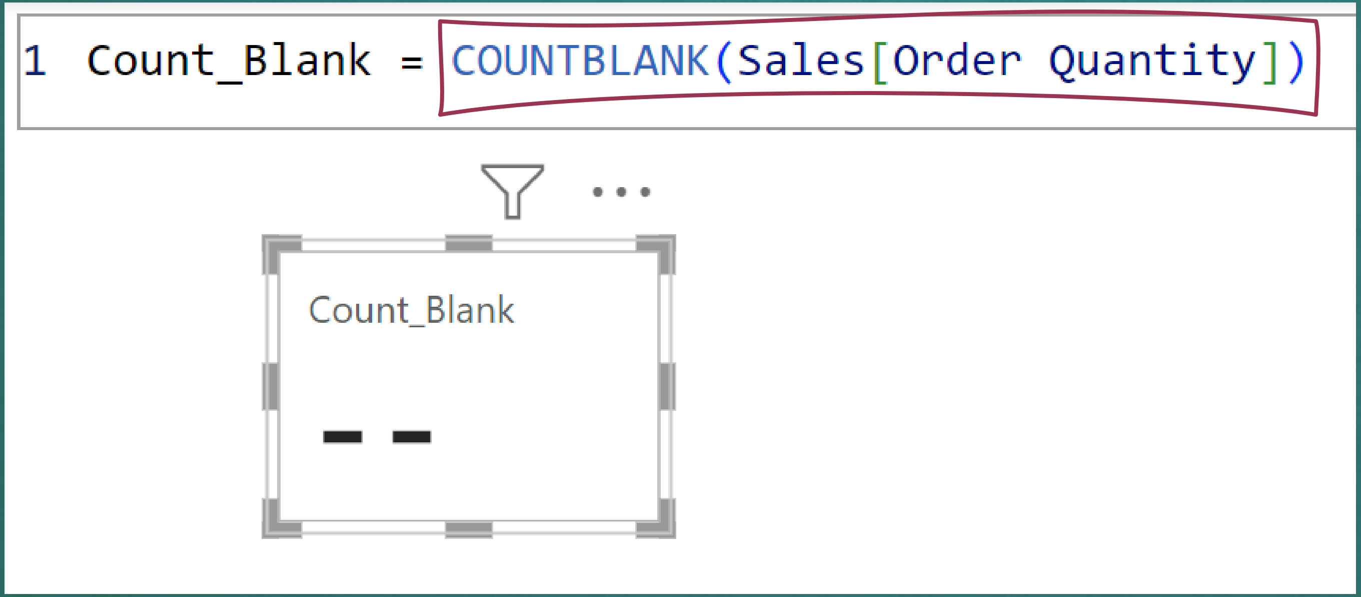
Count_rows

75K

Karthimeena

COUNTBLANK

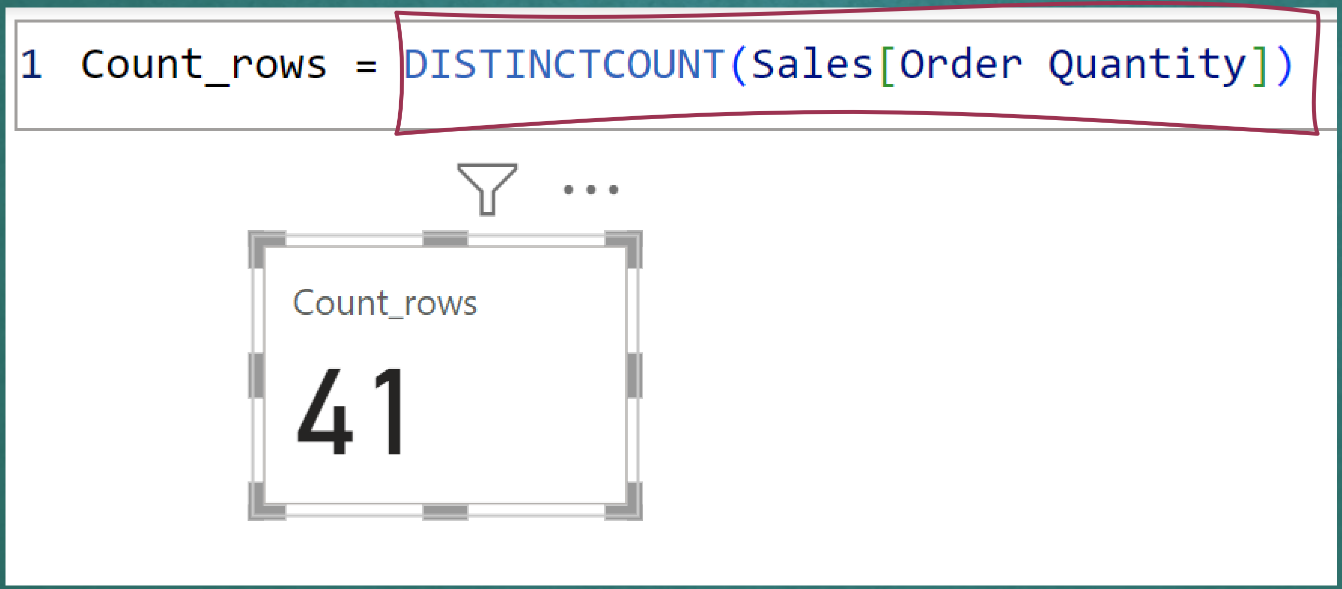
- COUNTBLANK() function counts the number of rows with BLANK value.



```
Count_Blank = COUNTBLANK(Sales[Order Quantity])
```

DISTINCTCOUNT

- This function counts the number of distinct values in a table.



```
Count_rows = DISTINCTCOUNT(Sales[Order Quantity])
```

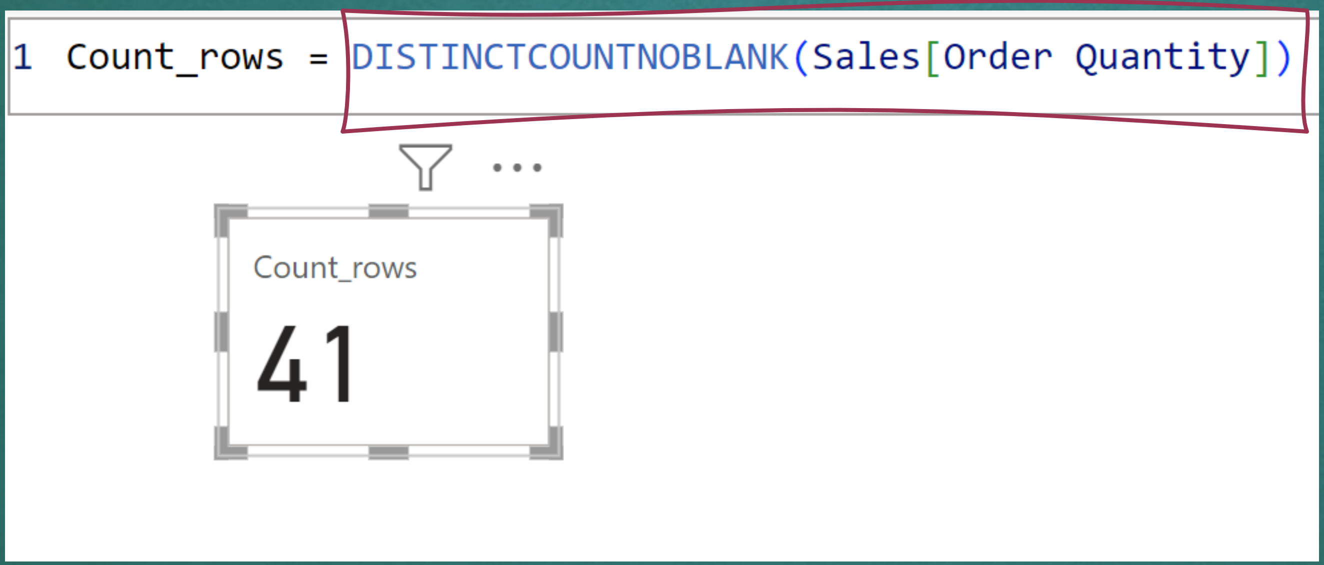
Count_rows

41

Karthimeena

DISTINCTCOUNTNOBLANK

- This function counts the number of distinct values in a table and does not include blank values if they occurs.



```
Count_rows = DISTINCTCOUNTNOBLANK(Sales[Order Quantity])
```

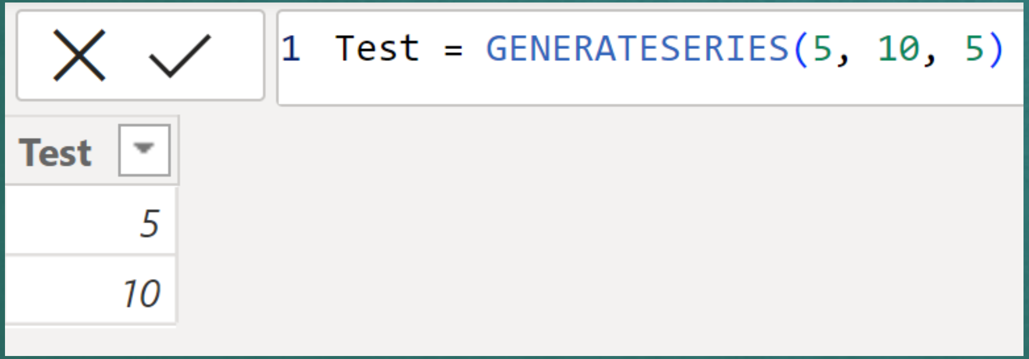
Count_rows

41

Karthimeena

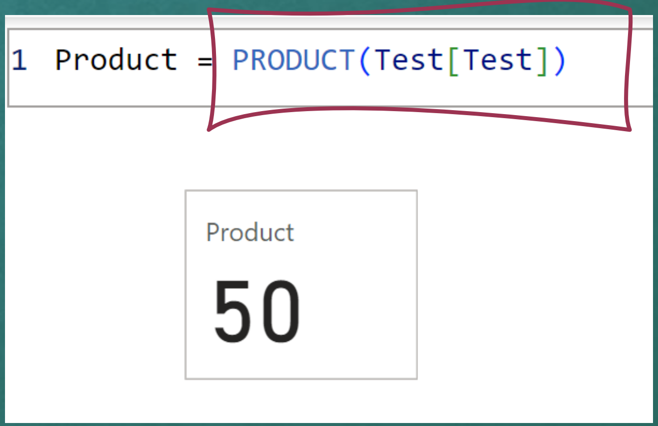
PRODUCT

- Product() used to calculate the product of numbers in the given column.
- It accepts column as a parameter and returns a numeric values.
- It counts only numbers in the column, it ignores blank, text and logical values.



```
Test = GENERATESERIES(5, 10, 5)
```

Test
5
10



```
Product = PRODUCT(Test[Test])
```

Product

50

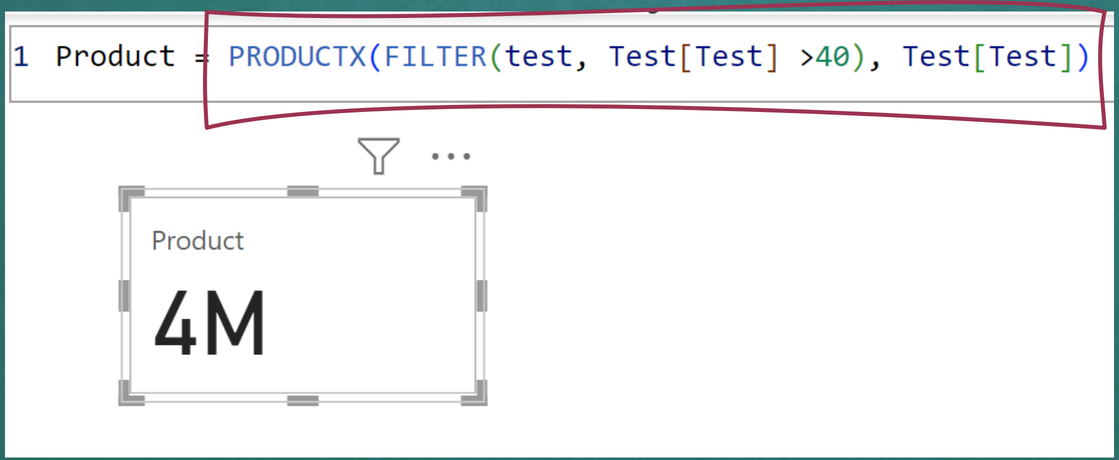
Karthimeena

PRODUCTX

- This returns the product of evaluated expression for each for in a table.
- It accepts table as first parameter, expression as second parameter.
- It counts only numbers in the column, it ignores blank, text and logical values.

PRODUCTX

- Below code returns the product of greater than the value 40 in a table.



```
Product = PRODUCTX(FILTER(test, Test[Test] >40), Test[Test])
```

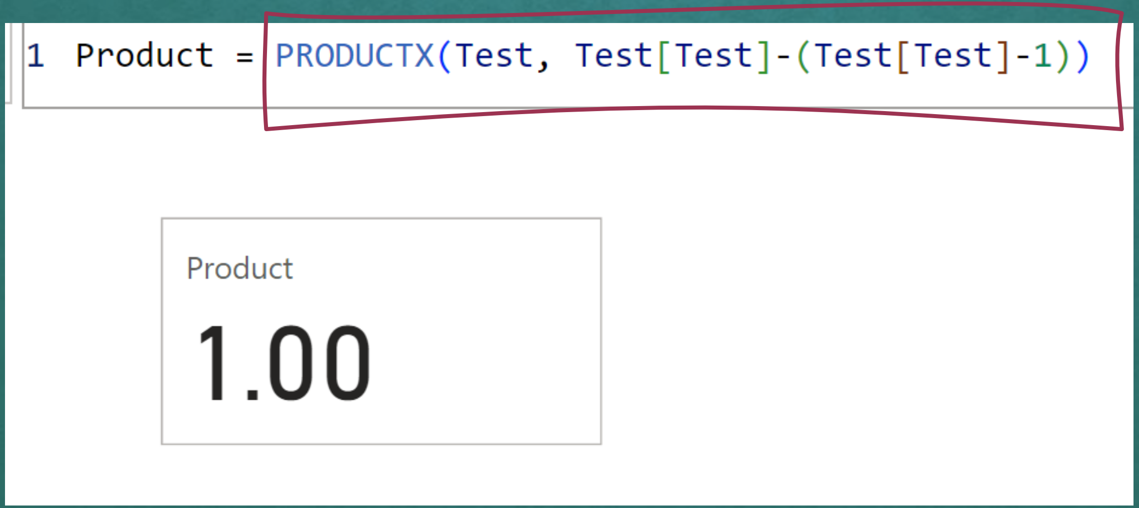
Product

4M

Karthimeena

PRODUCTX

- Below code evaluates the expression and find the product of all rows in a table



```
Product = PRODUCTX(Test, Test[Test]-(Test[Test]-1))
```

Product

1.00

Karthimeena