DATE : 20.05.2024

DT/NT : NT

LESSON : POWER BI

SUBJECT: DAX FUNCTIONS

BATCH : 247-253

TECHPRO EDUCATION

techproeducation.com

+1 (585) 304 29 59

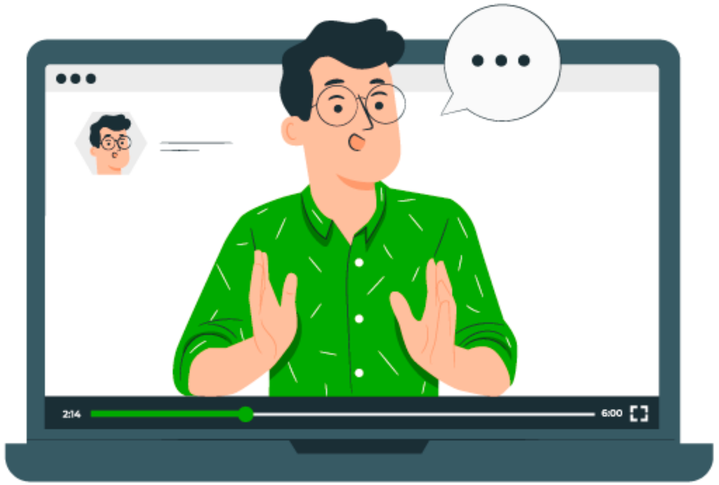
Power BI

POWER BI – DAX

Data Science Program

Session-10

Power BI Lesson Content



Bugün ne öğreneceğiz?

Power BI

- DAX (Data Analysis Expressions)
- DAX Formula Syntax
- DAX Functions
- Calculated Columns
- Measures
- Aggregations DAX Functions

DAX (Data Analysis Expressions) nedir?

- **Data Analysis Expressions (DAX);** Power BI'da kullanılan bir formül dilidir.
- Power BI raporlarında veri modellemesi ve analizi için kullanılan bir araçtır.
- DAX, tablo ve sütunlarla çalışarak veri analizi, hesaplamalar, ölçümler ve filtrelemeler gibi işlemleri gerçekleştirmek için kullanılır.
- DAX formülleri, tablolar arasındaki ilişkileri kullanarak verileri işleyebilir ve sonuçları hesaplayabilir.
- Modelinizde zaten bulunan verilerden yeni bilgiler oluşturmanıza yardımcı olabilir.

DAX & M LANGUAGE

	DAX	M
Why?	Calculations and analysis	Transformation and connections
Where?	Used in Power BI Desktop.	Used in Power Query Editor.
What?	Returns a single value, column, or table. Can make calculations using fields from many tables?	Control's shape of data when loaded into Power BI. Transforms one table at a time or combine tables.

Calculated Columns

- Veri modelindeki tablolara yeni, formül tabanlı sütunlar eklemenizi sağlar.
- Mevcut sütunlar veya diğer hesaplanmış sütunlardan değerleri alarak yeni bir değer üretir.
- Bu değerler, veri modelinin "Data" görünümünde tablolar içinde görünür.
- Statik veya sabit değerler için kullanılır. Agregasyon işlemleri (toplama, sayma vb.) için measures kullanılmalıdır

Calculated Columns

- "row context" yani satır bağlamınında kullanılırlar. Bu, bir hesaplanmış sütunun, tablodaki her bir satır için bağımsız değerler hesaplayabileceği anlamına gelir.
- Örneğin, bir tabloda bulunan "fiyat" ve "miktar" sütunlarından "gelir" adında bir hesaplanmış sütun oluşturabilirsiniz. Bu hesaplanmış sütun, her satır için fiyatı miktarla çarparak geliri hesaplar.

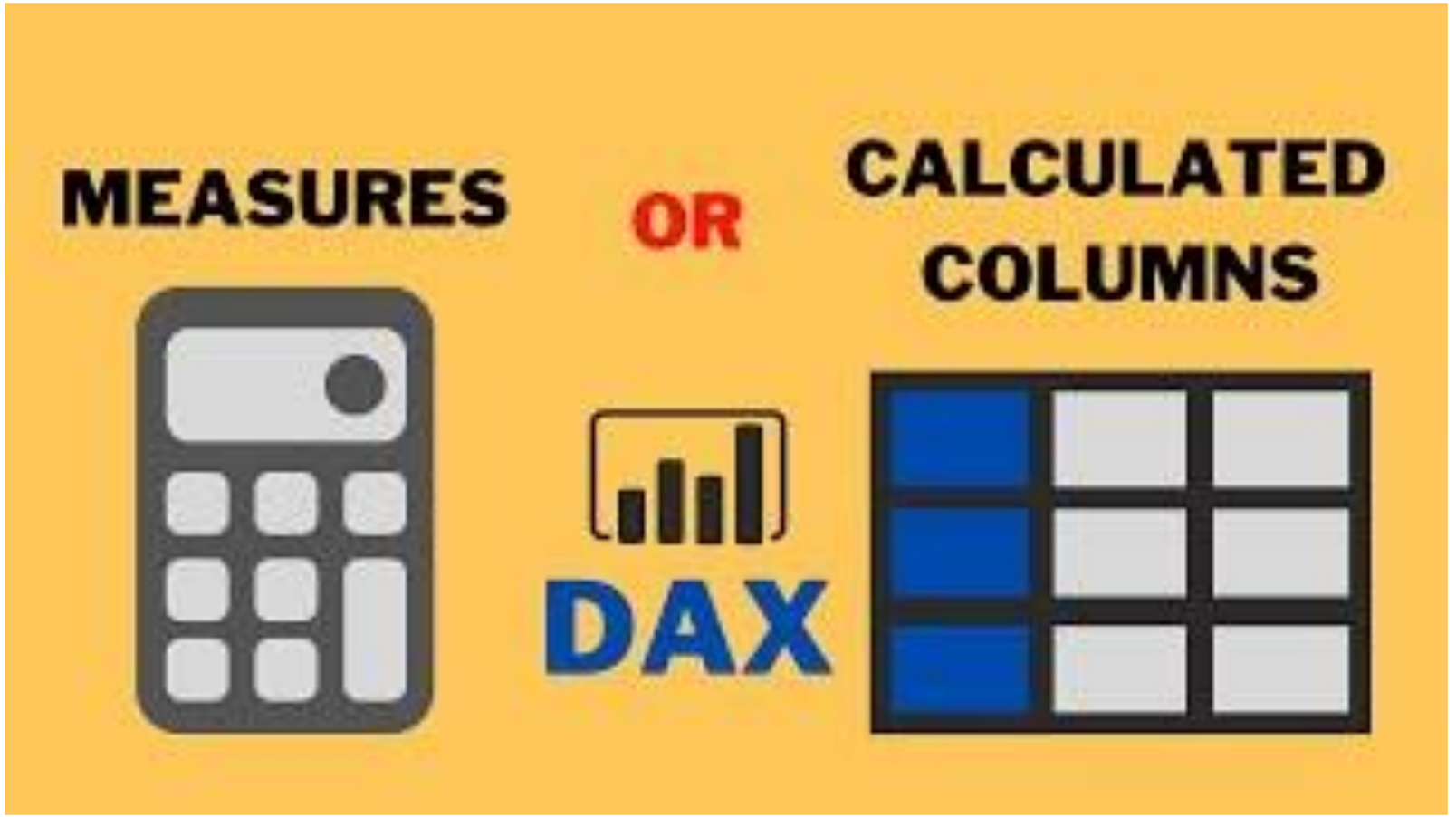
Measures

- Measures, raporlarınızda önemli metrikleri ve performans göstergelerini temsil etmek için kullanılır.
- Power BI raporlarında kullanılan ölçeklendirilebilir ve toplanabilir sayısal değerlerdir. Genellikle bir özetleme veya hesaplama işlemi sonucu elde edilir.
- Örneğin, satış geliri, toplam satış miktarı, ortalama fiyat, kar marjı gibi metrikler measures olarak tanımlanabilir.
- Ölçüler, tablolar içinde doğrudan görünmez; yalnızca bir görselleştirmede (örneğin bir grafik, tablo veya matris) “görülebilir”.

Measures

- Ölçüler, " filter context " bağlamında değerlendirilir ve etrafındaki alanlar veya filtreler değiştiğinde yeniden hesaplanır. Bu, ölçülerin raporun farklı bölümleri arasında etkileşimli olabileceği ve farklı dilimleyiciler veya filtreler uygulandığında farklı sonuçlar verebileceği anlamına gelir.

- Tek bir satırın cevap veremediği veya birden fazla satırdaki değerlerin toplanması gerektiğinde kullanılması gerekir. Bu, ölçülerin toplama, ortalama, maksimum, minimum ve diğer agregat hesaplamalar için kullanılması gerektiğini gösterir.



Calculated Columns & Measures

Calculated Columns

- Değerler, tablonun her bir satırındaki bilgilere dayanarak hesaplanır, bu da "satır bağlamı" (row context) olarak adlandırılır.
- Her bir satıra statik değerler ekler ve bu değerler veri modelinde saklanır, bu da dosya boyutunu artırır.
- Veri kaynağı yenilendiğinde veya temel sütunlarda değişiklik yapıldığında yeniden hesaplanırlar.
- Raporlarda verileri filtrelemek için çoğunlukla kullanılırlar.
- Hesaplanmış sütunlar tablolar içinde "yaşar", yani veri modelinin "Data" görünümünde görsel olarak gözlemlenebilirler.

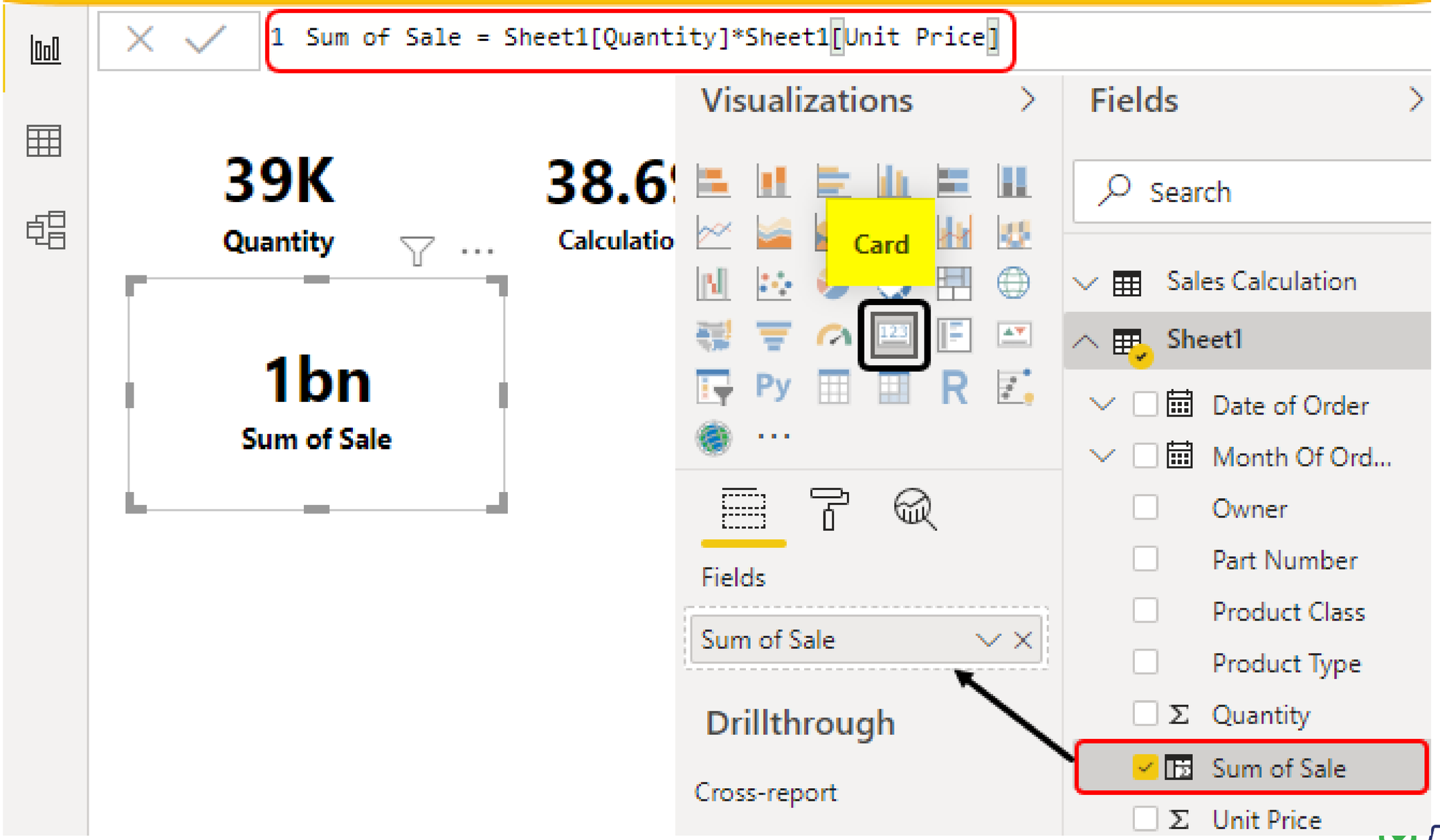
Measures

- Değerler, rapordaki herhangi bir filtreden gelen bilgilere dayanarak hesaplanır, bu da "filtre bağlamı" (filter context) olarak adlandırılır.
- Tabloların kendilerinde yeni veri oluşturmazlar, bu nedenle dosya boyutunu artırmazlar.
- Rapor içerisindeki filtrelerdeki herhangi bir değişikliğe yanıt olarak yeniden hesaplanırlar.
- Rapor görsellerinde değerleri toplamak için çoğunlukla kullanılırlar.
- Ölçüler görsellerde "yaşar", yani sadece rapor görünümündeki çizelgelerde veya diğer görselleştirmelerde gösterilirler.

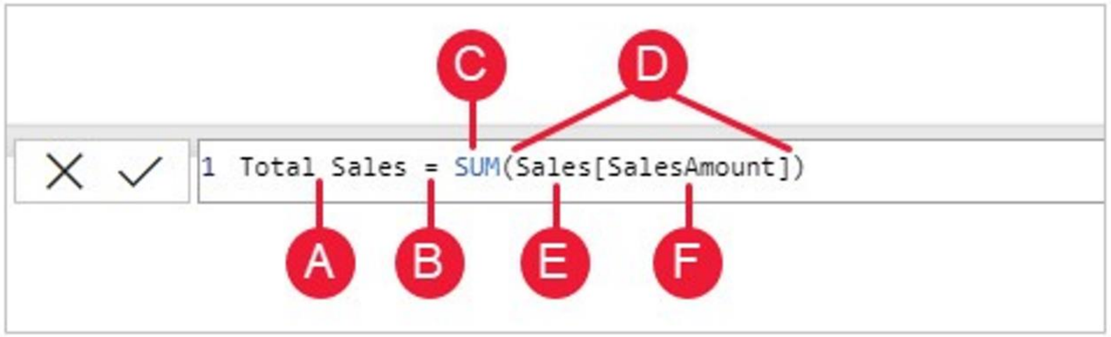
QUICK MEASURES

- Kullanıcıların yaygın hesaplama ihtiyaçlarını karşılamak için hızlı ve kolay bir şekilde DAX formülleri oluşturmalarına olanak tanıyan bir araçtır.
- Önceden oluşturulmuş şablonları veya doğal dil sorgularını kullanarak formüller oluşturmanızı sağlar. Bu, sık kullanılan hesaplamaları (ağırlıklı ortalamalar, yüzde farklar, zaman zekası işlemleri vb.) hızlı bir şekilde yapmanıza yardımcı olur.
- Başlangıç seviyesindeki kullanıcılar için veya daha karmaşık formüller oluşturmayı öğrenirken faydalı bir araç olabilir.

Power BI DAX


1 Sum of Sale = Sheet1[Quantity]*Sheet1[Unit Price]
39K
Quantity
38.6
Calculatio
1bn
Sum of Sale
Visualizations
Card
Fields
Sum of Sale
Drillthrough
Cross-report
Fields
Search
Sales Calculation
Sheet1
Date of Order
Month Of Ord...
Owner
Part Number
Product Class
Product Type
Quantity
Sum of Sale
Unit Price

DAX Syntax



Bu formül, aşağıdaki söz dizimi öğelerini içerir:

A. Ölçü adı: **Total Sales**.

B. Formülün başlangıcını gösteren eşittir işareti işleci (=). Hesaplama gerçekleştirildiğinde bir sonuç döndürür.

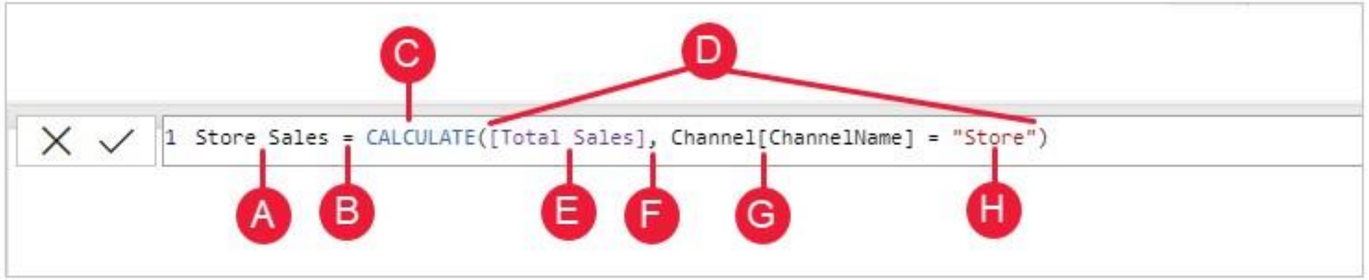
C. **Sales[SalesAmount]** sütunundaki tüm sayıları toplayan DAX işlevi **SUM**.

D. Bir veya daha fazla bağımsız değişken içeren ifadeyi içine alan ayraçlar (). Çoğu işlev için en az bir bağımsız değişken gerekir. Bağımsız değişken, bir işleve değer geçirir.

E. Başvurulan tablo: **Sales**.

F. Sales tablosunda başvurulan sütun: **[SalesAmount]**. Bu bağımsız değişken ile SUM işlevi, bir SUM oluşturmak için hangi sütunların toplanacağını belirtir.

DAX Syntax



Bu formülü daha iyi anlamak için tıpkı diğerlerinde olduğu gibi bu formülü de parçalara ayırarak inceleyebiliriz.

Bu formül, aşağıdaki söz dizimi öğelerini içerir:

A. Ölçü adı: **Store Sales**.

B. Formülün başlangıcını gösteren eşittir işareti işleci (=).

C. Belirtilen filtrelere göre değiştirilen bir bağlamda bir ifadeyi bağımsız değişken olarak değerlendiren **CALCULATE** işlevi.

D. Bir veya daha fazla bağımsız değişken içeren bir ifadeyi çevreleyen ayraçlar ().

E. Aynı tabloda bir ifade olarak bulunan **[Total Sales]**. Total Sales ölçüsün şu formüle sahiptir: =SUM(Sales[SalesAmount]).

F. İlk ifade bağımsız değişkenini filtre bağımsız değişkeninden ayıran virgül (,).

G. Başvurulan sütunun tam adı: **Channel[ChannelName]**. Bu, bizim satır bağlamımızdır. Bu sütundaki her bir satır Store veya Online gibi bir kanalı belirtir.

H. Filtre olarak kullanılan belirli değer: **Store**. Bu, bizim filtre bağlamımızdır.

Bu formül, filtre olarak Channel[ChannelName] sütunundaki yalnızca *Store* değerini içeren satırlar için yalnızca Total Sales ölçüsüyle tanımlanan satışların hesaplanmasını sağlar.

DAX OPERATORS

Arithmetic Operator	Meaning	Example
+	Addition	2 + 7
-	Subtraction	5 – 3
*	Multiplication	2 * 6
/	Division	4 / 2
^	Exponent	2 ^ 5

Comparison Operator	Meaning	Example
=	Equal to	[City]=“Boston”
>	Greater than	[Quantity]>10
<	Less than	[Quantity]<10
>=	Greater than or equal to	[Unit Price]>=2.5
<=	Less than or equal to	[Unit Price]<=2.5
<>	Not equal to	[Country]<>”Mexico”

Pay attention to these!

Text/Logical Operator	Meaning	Example
&	Concatenates two values to produce one text string	[City] & “ “ & [State]
&&	Create an AND condition between two logical expressions	([State]=“MA”) && ([Quantity]>10)
|| *(double pipe)*	Create an OR condition between two logical expressions	([State]=“MA”) || ([State]=“CT”)
IN	Creates a logical OR condition based on a given list (using curly brackets)	‘Store Lookup’[State] IN { “MA”, “CT”, “NY” }

COMMON FUNCTION CATEGORIES

MATH & STATS Functions

*Functions used for **aggregation** or iterative, row-level calculations*

Common Examples:

- SUM
- AVERAGE
- MAX/MIN
- DIVIDE
- COUNT/COUNTA
- COUNTROWS
- DISTINCTCOUNT

Iterator Functions:

- SUMX
- AVERAGEX
- MAXX/MINX
- RANKX
- COUNTX

LOGICAL Functions

*Functions that use **conditional expressions** (IF/THEN statements)*

Common Examples:

- IF
- IFERROR
- AND
- OR
- NOT
- SWITCH
- TRUE
- FALSE

TEXT Functions

*Functions used to manipulate **text strings** or **value formats***

Common Examples:

- CONCATENATE
- COMBINEVALUES
- FORMAT
- LEFT/MID/RIGHT
- UPPER/LOWER
- LEN
- SEARCH/FIND
- REPLACE
- SUBSTITUTE
- TRIM

FILTER Functions

*Functions used to **manipulate table** and **filter contexts***

Common Examples:

- CALCULATE
- FILTER
- ALL
- ALLEXCEPT
- ALLSELECTED
- KEEPFILTERS
- REMOVEFILTERS
- SELECTEDVALUE

TABLE Functions

*Functions that **create** or **manipulate tables** and output tables vs. scalar values*

Common Examples:

- SUMMARIZE
- ADDCOLUMNS
- GENERATESERIES
- DISTINCT
- VALUES
- UNION
- INTERSECT
- TOPN

DATE & TIME Functions

*Functions used to manipulate **date & time values** or handle time intelligence calculations*

Common Examples:

- DATE
- DATEDIFF
- YEARFRAC
- YEAR/MONTH
- DAY/HOUR
- TODAY/NOW
- WEEKDAY
- WEEKNUM
- NETWORKDAYS

Time Intelligence:

- DATESYTD
- DATESMTD
- DATEADD
- DATESBETWEEN

RELATIONSHIP Functions

*Functions used to **manage & modify table relationships***

Common Examples:

- RELATED
- RELATEDTABLE
- CROSSFILTER
- USERELATIONSHIP

BASIC MATH & STATS FUNCTIONS

SUM	Evaluates the sum of a column	=**SUM**(ColumnName)
AVERAGE	Returns the average (arithmetic mean) of all the numbers in a column	=**AVERAGE**(ColumnName)
MAX	Returns the largest value in a column or between two scalar expressions	=**MAX**(ColumnNameOrScalar1, *[Scalar2]*)
MIN	Returns the smallest value in a column or between two scalar expressions	=**MIN**(ColumnNameOrScalar1, *[Scalar2]*)
DIVIDE	Performs division and returns the alternate result (or blank) if DIV/0	=**DIVIDE**(Numerator, Denominator, *[AlternateResult]*)

1 SUM:

Returns the sum of all the numbers in a column.

Syntax: *SUM(ColumnName)*

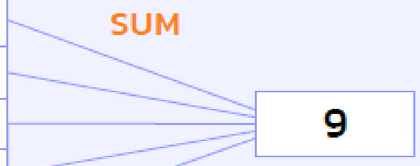
Example: Calculate the total quantity sold.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Quantity
2
1
3
1
2



Qty Sold = SUM(Sales[Quantity])

Qty Sold = SUM(2+1+3+1+2)

Qty Sold = 9

2 AVERAGE:

Returns the average (arithmetic mean) of all the numbers in a column.

Syntax: *AVERAGE(ColumnName)*

Example: Calculate the average price of transactions.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Price
500
800
450
900
850

AVERAGE → 700

AVG Price= AVERAGE(Sales[Price])

AVG Price= (500+800+450+900+850)/5

AVG Price= 700

3 MIN:

Returns the smallest value in a column, or between two scalar expressions.

Syntax: *MIN(ColumnName)* or *MIN(Expression1, Expression2)*

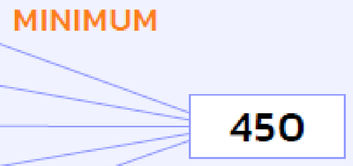
Example: Find the smallest price.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Price
500
800
450
900
850



MIN Price=
MIN(Sales[Price])

MIN Price=
MIN(500,800,450,900,850)

MIN Price= 450

4 MAX:

Returns the largest value in a column, or between two scalar expressions.

Syntax: *MAX(ColumnName)* or *MAX(Expression1, Expression2)*

Example: Find the largest price.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Price
500
800
450
900
850

MAXIMUM → 900

MAX Price=
MAX(Sales[Price])

MAX Price=
MAX(500,800,450,900,850)

MIN Price= 900

SUMX:

Returns the sum of an expression evaluated for each row in a table.

Syntax: ***SUMX(Table, Expression)***

Example: Calculate the total sales value (Quantity * Price for each transaction).

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Total Price=
SUMX(Sales,Sales[Quantity]*Sales[Price])

Total Price=
2*500+1*800+3*450+1*900+850

Total Price=
100+800+1350+900+1700

Total Price=5750

5 CALCULATE:

Modifies the filter context for a given expression.

Syntax: *CALCULATE(Expression, [Filter1, Filter2,...])*

Example: Calculate the total quantity sold for the 'North' region.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
3	A	3	400	North

North QTY= CALCULATE(SUM(Sales[Quantity]), Sales[Region]="North")

North QTY= 2+3

North QTY= 5

6 FILTER:

Returns a table that includes only the rows that meet a certain condition.

Syntax: ***FILTER(Expression,Filter)***

Example: Filters transactions over North region.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

North QTY=
FILTER(Sales,Sales[Region]="North")

Explanation:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
3	A	3	400	North

7 ALL:

Returns all rows in a table or all values in a column ignoring any filters that might have been applied.

Syntax: *ALL(TableName or ColumnName, [Column1,...])*

Example: Calculate the total quantity ignoring the Region filter.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

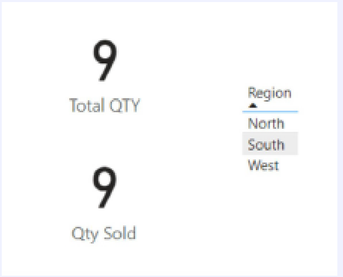
Explanation:

Total QTY=
CALCULATE(SUM(Sales[Quantity]), ALL(Region)

Total QTY=2+1+3+1+2

Total QTY=9

Without Filters:



Qty Sold=SUM(Sales[Quantity])

Qty Sold=SUM(2+1+3+1+2)

Qty Sold=9

With Filters:

8 ALLEXCEPT:

Removes all context filters in the table except filters that have been applied to the specified columns.

Syntax: *ALLEXCEPT(TableName, Column1,[Column2,...])*

Example: Calculate the total quantity ignoring all filters except the Region filter.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

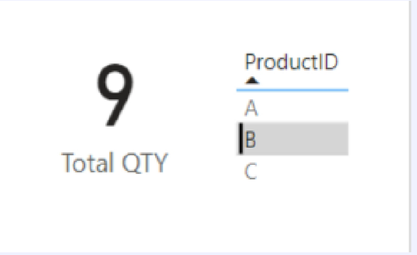
Explanation:

Total QTY=
CALCULATE(SUM(Sales[Quantity]), ALLEXCEPT(Region)

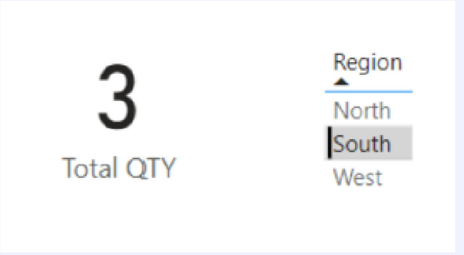
Total QTY=2+1+3+1+2

Total QTY=9

With Other Filters:



With Region Filter:



9 DISTINCT:

Returns a table containing only distinct rows.

Syntax: *DISTINCT(TableName)*

Returns a column of unique values.

Syntax: *DISTINCT(ColumnName)*

Example: List unique product IDs sold.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Products=DISTINCT(Sales[ProductID])

Product ID
A
B
C

MATH AND TRIG FUNCTIONS:

These are functions in DAX that allow for the execution of mathematical and trigonometric operations on data.

10 ABS:

Returns the absolute value of a number.

Syntax: *ABS(Number)*

Example: ABS(10-15)
5

11 DIVIDE:

Performs division and returns an alternate result or BLANK on division by 0.

Syntax: *DIVIDE(Numerator, Denominator, [AlternateResult])*

Example: = DIVIDE(8,2,0)
= 4

Example: DIVIDE(3,0,0)
= 0

12 IF:

Checks a condition, and returns one value if True, and another value if False.

Syntax: *IF(LogicTest, ResultIfTrue, [ResultIfFalse])*

Example: Categorize transactions as 'High' or 'Low' based on price.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

Category = IF(Sales[Price] >= 800, "High", "Low")

Price	Applying condition	Category
500	→	Low
800	→	High
450	→	Low
900	→	High
850	→	High

13 SWITCH:

Evaluates an expression against a list of values and returns the result corresponding to the first matching value.

Syntax: *SWITCH(Expression, Value1, Result1, Value2, Result2, ..., [DefaultResult])*

Example: Categorize transactions as 'High Price' or 'Medium Price' or 'Low Price' based on price.

'Sales' Table:

Transaction ID	Product ID	Quantity	Price	Region
1	A	2	500	North
2	B	1	800	South
3	A	3	450	North
4	C	1	900	West
5	B	2	850	South

Explanation:

```
Price Category = SWITCH(TRUE(),
    Sales[Price] >= 800, "High Price",
    Sales[Price] >= 500 && Sales[Price] < 800, "Medium Price",
    Sales[Price] < 500, "Low Price",
    "Undefined" // Default case if no other conditions are met
)
```

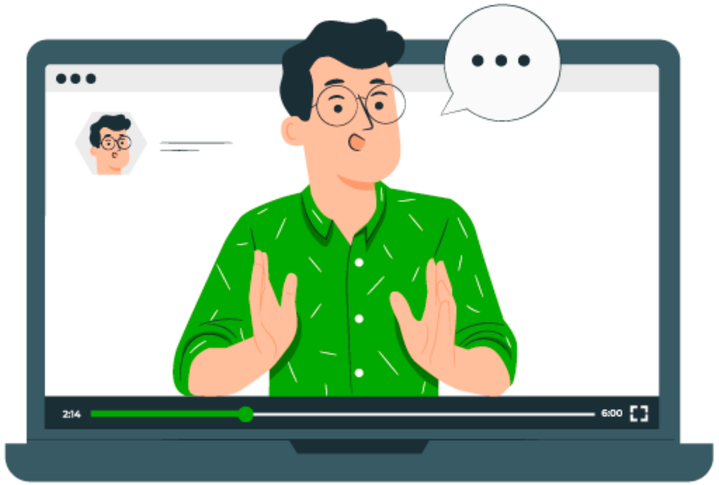
Price	Applying condition	Price Category
500	→	Medium Price
800	→	High Price
450	→	Low Price
900	→	High Price
850	→	High Price

NumPy

Sorunuz var mı ?

TECHPRO EDUCATION

Power BI Lesson Content



Sonraki derste ne öğreneceğiz?

Power BI

- Logical DAX Functions
- Filter DAX Functions
- Text DAX Functions
- Time Intelligence DAX Functions
- Date & Time DAX Functions

Tea break...

10:00