

زبان DAX

Data Analysis eXpression

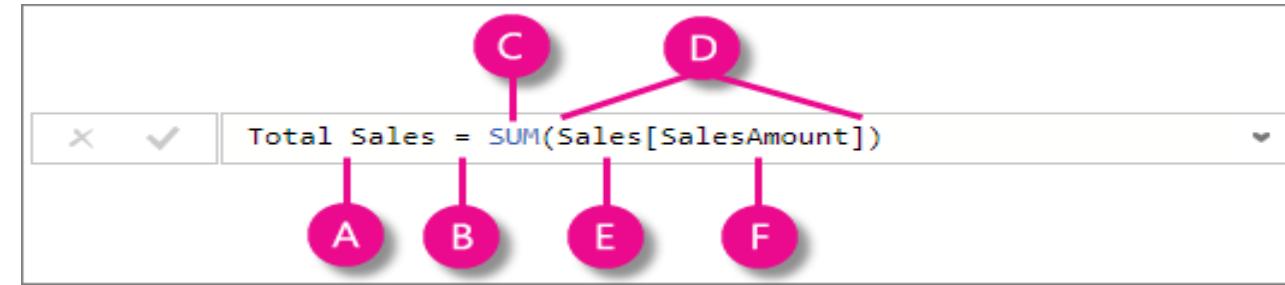
DAX

- زبان DAX یک کتابخانه از توابع و اپراتورهایی است که می‌تواند برای ساخت فرمول و عبارات ترکیب شوند که در نهایت می‌تواند یک یا چندین مقدار را برگرداند.
- زبان DAX زبان برنامه نویسی نیست.
- DAX به شما کمک می‌کند اطلاعات جدیدی را از داده‌ای موجود در مدل داده‌ای خود ایجاد کنید.
- فرمول‌های DAX شما را قادر به انجام مدل سازی داده‌ها، تجزیه و تحلیل داده‌ها و استفاده از نتایج برای گزارش‌گیری و تصمیم‌گیری می‌کند.

DAX:

- Data Analysis Expressions (DAX) is a library of functions and operators that can be combined to build formulas and expressions in Power BI, Analysis Services, and Power Pivot in Excel data models.
- Once a physical table exists in a Power BI, DAX cannot add, change, or remove data from that table.
- Data can only be filtered or queried using DAX functions.
- DAX formulas are used in measures, calculated columns, calculated tables, and row-level security.
- DAX formulas include functions, operators, and values to perform advanced calculations and queries on data in related tables and columns in tabular data models.

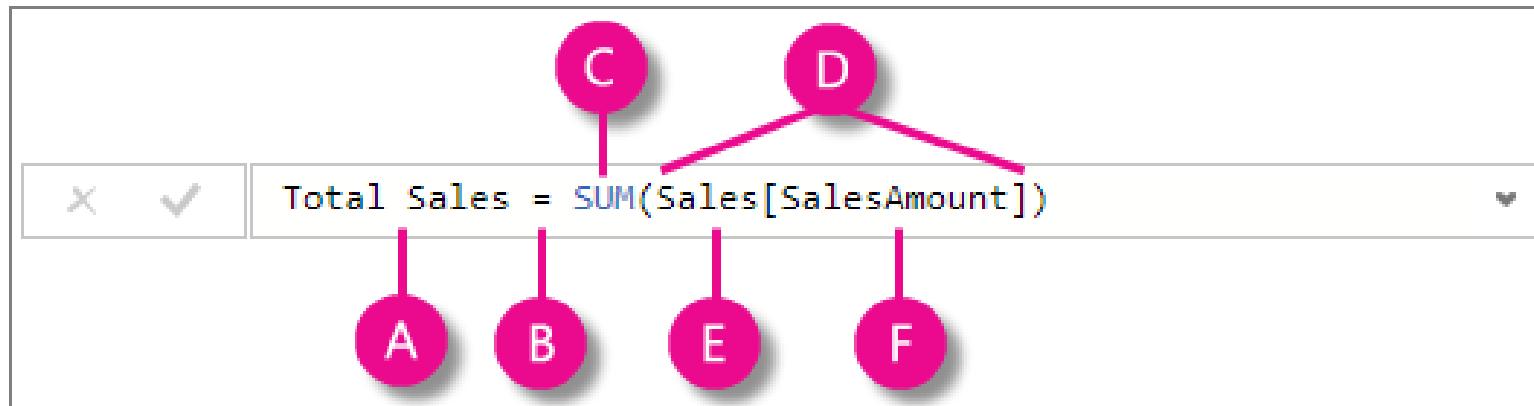
Parts of DAX Formula



- A. The measure name, **Total Sales**.
- B. The equals sign operator (=), which indicates the beginning of the formula. When calculated, it will return a result.
- C. The DAX function **SUM**, which adds up all of the numbers in the **Sales[SalesAmount]** column. You'll learn more about functions later.
- D. Parenthesis (), which surround an expression that contains one or more arguments. All functions require at least one argument. An argument passes a value to a function.
- E. The referenced table, **Sales**.
- F. The referenced column, **[SalesAmount]**, in the Sales table. With this argument, the SUM function knows on which column to aggregate a SUM.

فرمت زبان DAX

- .A. عنوان شاخص ، قلم آماری (عنوان دلخواه)
- .B. عملگر (=) نشانگر شروع فرمول محاسبه است.
- .C. تابع Sum جمع ستونی فیلد SalesAmount می باشد
- .D. نام جدول به همراه فیلد
- .E. نام جدول
- .F. ستون مورد نظر (ستون داخل براکت)



For the measure named Total Sales, calculate (=) the SUM of values in the [SalesAmount] column in the Sales table.

نوشتن توضیحات (Comments) در فرمول:

```
Measure = DATEDIFF(  
    -- This is the first argument  
    TODAY() ,  
    // This is the second argument  
    DATE(2017,12,24),  
    /* This is the final argument */  
    DAY  
)
```

استفاده از :Quotation marks

توضیح	مثال	شی
در عنوان جدول از کاراکتر Special یا کاراکترهای Space استفاده نشده است	Sales	جدول
در عنوان جدول از کاراکتر Special یا کاراکترهای Space استفاده شده است برای این منظور نام جدول داخل تک کووییشن قرار می گیرد	'Canada Sales'	جدول
برای نوشتن کامل فیلد بهتر است نام جدول در ادامه داخل برآکت فیلد ذکر شود	Sales[Amount]	عنوان کامل ستون
	'Canada Sales'[Amount]	عنوان کامل ستون

Types of Operators:

- **Arithmetic operators**
- **Comparison operators**
- **Text concatenation operator**
- **Logical operators**

Arithmetic operators

Arithmetic operator	Meaning	Example
<code>+</code> (plus sign)	Addition	$3+3$
<code>-</code> (minus sign)	Subtraction or sign	$3-1-1$
<code>*</code> (asterisk)	Multiplication	$3*3$
<code>/</code> (forward slash)	Division	$3/3$
<code>^</code> (caret)	Exponentiation	16^4

Comparison operators

Comparison operator	Meaning	Example
=	Equal to	[Region] = "USA"
==	Strict equal to	[Region] == "USA"
>	Greater than	[Sales Date] > "Jan 2009"
<	Less than	[Sales Date] < "Jan 1 2009"
>=	Greater than or equal to	[Amount] >= 20000
<=	Less than or equal to	[Amount] <= 100
<>	Not equal to	[Region] <> "USA"

All comparison operators except == treat BLANK as equal to number 0, empty string "", DATE(1899, 12, 30), or FALSE. As a result, [Column] = 0 will be true when the value of [Column] is either 0 or BLANK. In contrast, [Column] == 0 is true only when the value of [Column] is 0.

Text concatenation operator

Text operator	Meaning	Example
& (ampersand)	Connects, or concatenates, two values to produce one continuous text value	[Region] & ", " & [City]

Logical operators

Use logical operators (`&&`) and (`||`) to combine expressions to produce a single result.

Text operator	Meaning	Examples
<code>&&</code> (double ampersand)	Creates an AND condition between two expressions that each have a Boolean result. If both expressions return TRUE, the combination of the expressions also returns TRUE; otherwise the combination returns FALSE.	<code>([Region] = "France") && ([BikeBuyer] = "yes")</code>
<code> </code> (double pipe symbol)	Creates an OR condition between two logical expressions. If either expression returns TRUE, the result is TRUE; only when both expressions are FALSE is the result FALSE.	<code>(([Region] = "France") ([BikeBuyer] = "yes"))</code>
<code>IN</code>	Creates a logical OR condition between each row being compared to a table. Note: the table constructor syntax uses curly braces.	<code>'Product'[Color] IN { "Red", "Blue", "Black" }</code>

Calculation order

An expression evaluates the operators and values in a specific order. All expressions always begin with an equal sign (=). The equal sign indicates that the succeeding characters constitute an expression.

Following the equal sign are the elements to be calculated (the operands), which are separated by calculation operators. Expressions are always read from left to right, but the order in which the elements are grouped can be controlled to some degree by using parentheses.

Operator precedence

If you combine several operators in a single formula, the operations are ordered according to the following table. If the operators have equal precedence value, they are ordered from left to right. For example, if an expression contains both a multiplication and division operator, they are evaluated in the order that they appear in the expression, from left to right.

Operator	Description
$^$	Exponentiation
$-$	Sign (as in -1)
$*$ and $/$	Multiplication and division
!	NOT (unary operator)
$+$ and $-$	Addition and subtraction
$&$	Connects two strings of text (concatenation)
$=, ==, <, >, <=, >=, <>$	Comparison

Example:

DAX	
=5+2*3	

DAX	
= $(5+2)*3$	

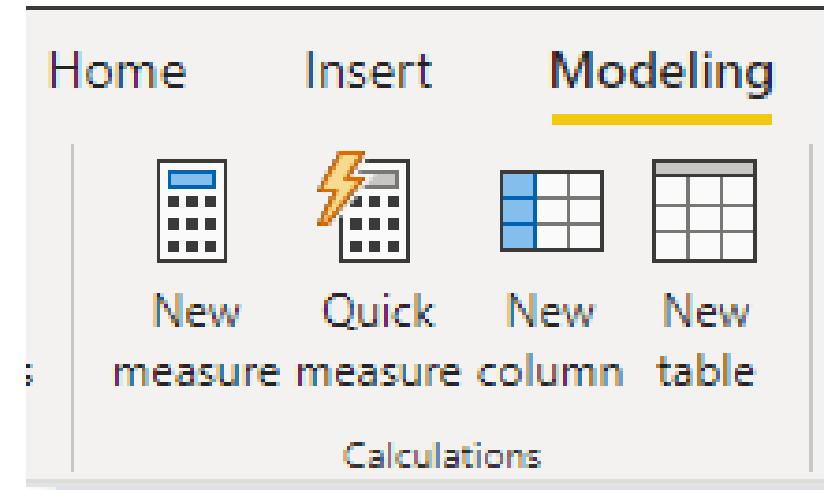
DAX	
= $(3 + 0.25)/(3 - 0.25)$	

DAX	
= -2^2	

DAX	
= $(-2)^2$	

Calculation in DAX:

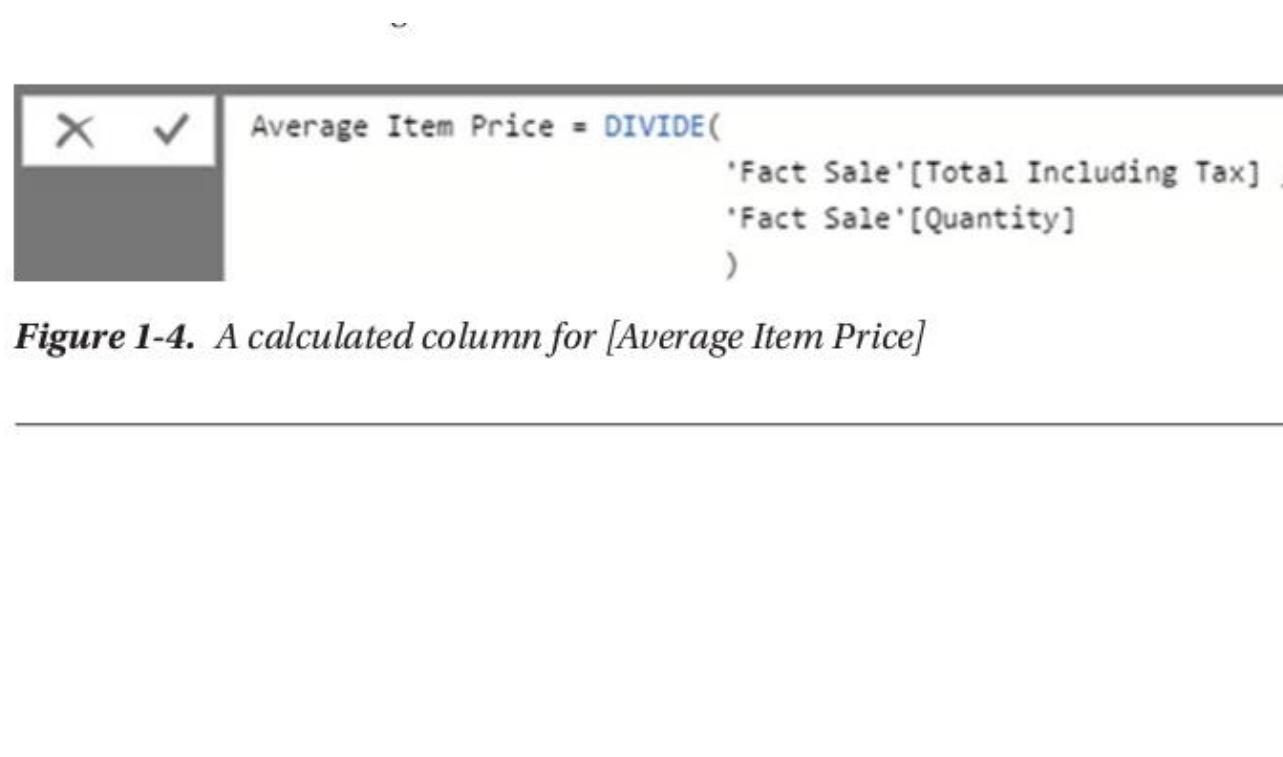
- Calculated columns
- Calculated measures
- Calculated tables



ستون Calculated Column

- در صورتی که بخواهیم سطر به سطر محاسبات انجام پذیرد از Calculated Column استفاده می شود.
- محاسبه و ذخیره می شود. (Pre calculated & Store).
- داده ها بعد از فیلتر کاربر محاسبه مجدد نمی شوند (Not Recalculated).
- در زمانی که منبع اطلاعات رفرش (بارگذاری مجدد) شود محاسبات مجدد انجام خواهد شد.
- فضای RAM را اشغال می کند، از فضای دیسک هم استفاده می کند.
- به عنوان فیلد ارتباطی بین جداول می توانید استفاده کنید.
- در حجم اطلاعات بالا کارایی داشبورد پایین می آید.

Calculated Column Example:



A screenshot of a Power BI interface showing a calculated column. On the left, there's a small toolbar with a 'X' and a checkmark. The main area displays the DAX formula for the calculated column:

```
Average Item Price = DIVIDE(  
    'Fact Sale'[Total Including Tax] ,  
    'Fact Sale'[Quantity]  
)
```

Figure 1-4. A calculated column for [Average Item Price]

Note This formula works without the table name preceding each column name; however, it is highly recommended that whenever you reference a column, you always include the table name. This makes it much easier to differentiate between columns and measures when you are debugging longer DAX queries.

Quantity	Total Including Tax	Average Item Price
90	1552.5	\$17.25
90	3001.5	\$33.35
90	3829.5	\$42.55
90	10246.5	\$113.85
90	3881.25	\$43.13
90	3881.25	\$43.13
90	2277	\$25.30
90	3001.5	\$33.35
90	4347	\$48.30
90	3415.5	\$37.95
90	2277	\$25.30
90	1552.5	\$17.25
..

Example:

Formula	Definition
= TODAY()	Inserts today's date in every row of a calculated column.
= 3	Inserts the value 3 in every row of a calculated column.
= [Column1] + [Column2]	Adds the values in the same row of [Column1] and [Column2] and puts the results in the calculated column of the same row.

فیلد محاسباتی Measure

- در صورتی که بخواهیم محاسبات سطری یا مجموعه از سطرها را داشته باشیم می توانیم از Measure استفاده کنیم.
- در Ram ذخیره نمی شود.
- در صورتی که فیلتری انجام شود داده هایش محاسبه مجدد(recalculate) می شود.
- قابلیت استفاده مجدد در قلم آماری دیگر را نیز دارد.
- نمی توانید به عنوان فیلد ارتباطی بین جداول استفاده کنید.

Calculated Measure Example:

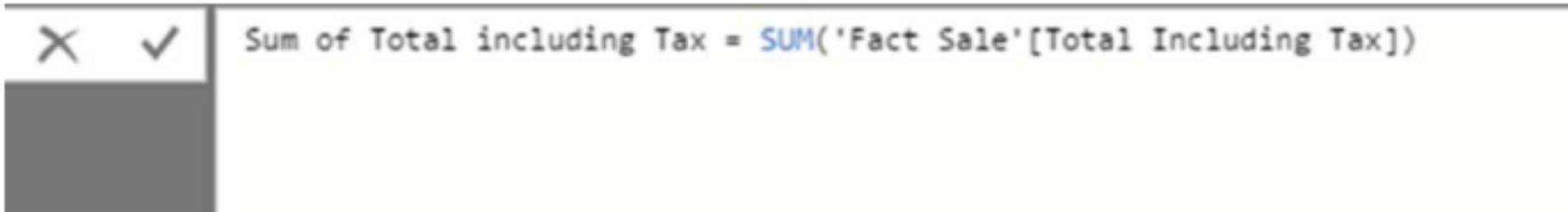


Figure 1-2. DAX for the first calculated measure. This calculation uses the SUM function to return a single value anywhere the calculated measure is used in the report. When dragged and dropped to the report canvas using a visual with no other fields or filters, the value should show as \$198,043,493.45.



مثال فیلد محاسباتی

```
1 Sales[SalesAmount] := Sales[Quantity] * Sales[Unit Price]
2 Sales[TotalProductCost] := Sales[Quantity] * Sales[Unit Cost]
3 Sales[GrossMargin] := Sales[SalesAmount] - Sales[TotalProductCost]
4 Sales[Gross Margin %] := DIVIDE ( SUM ( Sales[GrossMargin] ), SUM ( Sales[SalesAmount] ) )
```

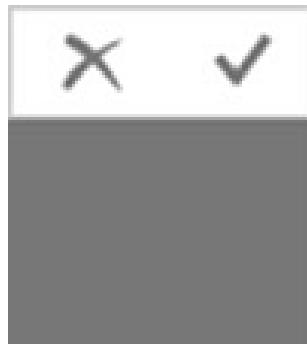
```
1 Sales[Gross Margin %] :=
2 DIVIDE (
3     SUMX ( Sales, Sales[Quantity] * Sales[Unit Price] )
4     - SUMX ( Sales, Sales[Quantity] * Sales[Unit Cost] ),
5     SUMX ( Sales, Sales[Quantity] * Sales[Unit Price] )
6 )
```

```
1 Sales[Gross Margin %] :=
2 VAR SalesAmount = SUMX ( Sales, Sales[Quantity] * Sales[Unit Price] )
3 VAR TotalProductCost = SUMX ( Sales, Sales[Quantity] * Sales[Unit Cost] )
4 VAR GrossMargin = SalesAmount - TotalProductCost
5 RETURN DIVIDE ( GrossMargin, SalesAmount )
```

جدول Table

- در صورتی که خروجی تابع یا اطلاعات بیش از یک فیلد باشد نوع خروجی اطلاعات باید جدول باشد .
- توابعی مثل Summarize، Group By و... نوع خروجی آنها جدول می باشد .

Calculated Tables Example:



```
Dates2 = FILTER(
    Dates,
    'Dates'[Calendar Year]=2016
)
```

Figure 1-7. Calculated table created using FILTER

DAX Function types:

- Date and time functions
- Filter functions
- Financial functions
- Information functions
- Logical functions
- Math and Trig functions
- Other functions
- Parent and Child functions
- Relationship functions
- Statistical functions
- Table manipulation functions
- Text functions
- Time intelligence functions

<https://docs.microsoft.com/en-us/dax/dax-function-reference>

DAX Functions:

- [Logical functions](#) - These functions return information about values in an expression. For example, the TRUE function lets you know whether an expression that you are evaluating returns a TRUE value.
- [Text functions](#) - With these functions, you can return part of a string, search for text within a string, or concatenate string values. Additional functions are for controlling the formats for dates, times, and numbers.
- [Date and time functions](#) - These functions in DAX are similar to date and time functions in Microsoft Excel. However, DAX functions are based on the datetime data types used by Microsoft SQL Server.
- [Time intelligence functions](#) - These functions help you create calculations that use built-in knowledge about calendars and dates. By using time and date ranges in combination with aggregations or calculations, you can build meaningful comparisons across comparable time periods for sales, inventory, and so on.
- [Math and Trig functions](#) - Mathematical functions in DAX are similar to Excel's mathematical and trigonometric functions. However, there are some differences in the numeric data types used by DAX functions.
- [Statistical functions](#) - These functions perform aggregations. In addition to creating sums and averages, or finding minimum and maximum values, in DAX you can also filter a column before aggregating or create aggregations based on related tables.
- [Filter functions](#) - These functions help you return specific data types, look up values in related tables, and filter by related values. Lookup functions work by using tables and relationships between them. Filtering functions let you manipulate data context to create dynamic calculations.

DAX Functions:

- Information functions - These functions look at a table or column provided as an argument to another function and tells you whether the value matches the expected type. For example, the ISERROR function returns TRUE if the value you reference contains an error.
- Relationship functions - These functions are for managing and utilizing relationships between tables. For example, you can specify a particular relationship to be used in a calculation.
- Table manipulation functions - These functions return a table or manipulate existing tables.
- Other functions - These functions perform unique actions that cannot be defined by any of the categories most other functions belong to.
- Parent and Child functions - These Data Analysis Expressions (DAX) functions help users manage data that is presented as a parent/child hierarchy in their data models.
- Financial functions - These functions are used in formulas that perform financial calculations, such as net present value and rate of return.

توابع **Logical Functions**

• توابع منطقی

ایجاد شرط ، استفاده از <>و<> و <>یا<> ،

استفاده از ... و Switch در این دسته قرار

- » [DAX functions](#)
- » [New DAX functions](#)
- » [Date and time functions](#)
- » [Time-intelligence functions](#)
- » [Filter functions](#)
- » [Information functions](#)
- » [Logical functions](#)
- » [Math and trig functions](#)
- » [Other functions](#)
- » [Parent and child functions](#)
- » [Statistical functions](#)
- » [Text functions](#)

دارد.

Logical Function

Function Description

<u>AND</u>	Checks whether both arguments are TRUE, and returns TRUE if both arguments are TRUE.
<u>COALESC</u>	Returns the first expression that does not evaluate to BLANK.
<u>FALSE</u>	Returns the logical value FALSE.
<u>IF</u>	Checks a condition, and returns one value when it's TRUE, otherwise it returns a second value.
<u>IFERROR</u>	Evaluates an expression and returns a specified value if the expression returns an error
<u>NOT</u>	Changes FALSE to TRUE, or TRUE to FALSE.
<u>OR</u>	Checks whether one of the arguments is TRUE to return TRUE.
<u>SWITCH</u>	Evaluates an expression against a list of values and returns one of multiple possible result expressions.
<u>TRUE</u>	Returns the logical value TRUE.

Logical Function

- IF :
 - برای ایجاد شرط استفاده می شود.

DAX

```
IF(logical_test,<value_if_true>, value_if_false)
```

DAX

```
=IF([Calls]<200,"low",IF([Calls]<300,"medium","high"))
```

Logical Function

- Switch : یکی دیگر از دستورات شرطی که همانند یک لیست عمل می کند و مقداری که برابر است را بر می گرداند دستور Switch می باشد

DAX

```
SWITCH(<expression>, <value>, <result>[, <value>, <result>]...[, <else>])
```

DAX

```
=SWITCH([Month], 1, "January", 2, "February", 3, "March", 4, "April"  
        , 5, "May", 6, "June", 7, "July", 8, "August"  
        , 9, "September", 10, "October", 11, "November", 12, "December"  
        , "Unknown month number" )
```

Logical Function

- AND :
 - این تابع بررسی می کند که هر دو عبارت صحیح می باشند یا خیر و اگر هر دو آرگومان صحیح باشد True و در غیر اینصورت False برمی گرداند.

DAX

```
AND(<logical1>,<logical2>)
```

DAX

```
-IF(AND(10 > 9, -10 < -1), "All true", "One or more false")
```

```
USA Gold Medal Count:= SUMX(  
    Results, IF(AND([Country] = "USA", [Medal] = "Gold") = TRUE(),1,0)  
)
```

Logical Function

- OR:

- این تابع بررسی می کند که آیا یکی از دو عبارت صحیح می باشند یا خیر و اگر هر یکی از آرگومان ها صحیح باشد True و در غیر اینصورت False برمی گرداند.

DAX

```
OR(<logical1>,<logical2>)
```

Logical Function

- IN: این تابع عمل OR را در یک لیست بررسی می کند.
-

DAX

IN

DAX

```
Filtered Sales:=CALCULATE (
    [Internet Total Sales], 'Product'[Color] IN { "Red", "Blue", "Black" }
)
```

Logical Function

- IFERROR: در صورتی که عبارت با خطا مواجه شود می توان شرطی را برای آن پیاده سازی کرد.

DAX

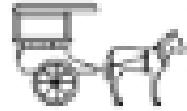
```
IFERROR(value, value_if_error)
```

DAX

```
=IFERROR(25/0,9999)
```

```
= IFERROR (5/0,"Div by zero") returns Div by zero.  
= IFERROR (5/1,"Div by zero") returns 5.
```

مثال ۱ - ایجاد Indicator از طریق ICON

vehicle	Grade	Indicator
bicycle	11	
car	15	
carriage	8	
motor	12	
Total	46	

توابع Text Functions

• توابع متنی

این توابع محاسباتی بر روی رشته انجام می دهند ، عملیاتی مثل جمع کردن دو رشته ، واکشی مقداری از رشته و ...

- » [DAX functions](#)
 - » [New DAX functions](#)
 - » [Date and time functions](#)
 - » [Time-intelligence functions](#)
 - » [Filter functions](#)
 - » [Information functions](#)
 - » [Logical functions](#)
 - » [Math and trig functions](#)
 - » [Other functions](#)
 - » [Parent and child functions](#)
 - » [Statistical functions](#)
 - » [Text functions](#)

Text Function

Function	Description
COMBINEVALUES	Joins two or more text strings into one text string.
CONCATENATE	Joins two text strings into one text string.
CONCATENATEX	Concatenates the result of an expression evaluated for each row in a table.
EXACT	Compares two text strings and returns TRUE if they are exactly the same, FALSE otherwise.
FIND	Returns the starting position of one text string within another text string.
FIXED	Rounds a number to the specified number of decimals and returns the result as text.
FORMAT	Converts a value to text according to the specified format.
LEFT	Returns the specified number of characters from the start of a text string.
LEN	Returns the number of characters in a text string.
LOWER	Converts all letters in a text string to lowercase.
MID	Returns a string of characters from the middle of a text string, given a starting position and length.
REPLACE	REPLACE replaces part of a text string, based on the number of characters you specify, with a different text string.
REPT	Repeats text a given number of times.
RIGHT	RIGHT returns the last character or characters in a text string, based on the number of characters you specify.
SEARCH	Returns the number of the character at which a specific character or text string is first found, reading left to right.
SUBSTITUTE	Replaces existing text with new text in a text string.
TRIM	Removes all spaces from text except for single spaces between words.
UNICHAR	Returns the Unicode character referenced by the numeric value.
UNICODE	Returns the numeric code corresponding to the first character of the text string.
UPPER	Converts a text string to all uppercase letters.
VALUE	Converts a text string that represents a number to a number.

Text Function

- Concatenate:
 - برای الحاق دو رشته می توان از تابع Concatenate استفاده کرد.
 - در صورتی که بخواهیم سطر به سطر یک جدول محاسبات الحاق انجام بدهیم و با یک جداگانه آنها را بهم متصل کنیم می توانیم از ConcatenateX استفاده کنیم.
- ConcatenateX:

DAX

```
CONCATENATE(<text1>, <text2>)
```

DAX

```
CONCATENATEX(<table>, <expression>, [delimiter])
```

Text Function

- Exact:
 - برسی می کند که آیا دو رشته با هم برابر هستند یا خیر

DAX

```
EXACT(<text1>,<text2>)
```

- Find:
 - در صورتی که رشته اول در رشته دوم وجود داشته باشد عدد اندیس مورد نظر را بر می گرداند

DAX

```
FIND(<find_text>,<within_text>[, <start_num>][, <NotFoundValue>])
```

DAX

```
=FIND("BMX","line of BMX racing goods")
```

Text Function

- Fixed: این تابع فرمت نمایشی اعداد را مشخص می کند و خروجی آن به صورت رشته می باشد
-

DAX

```
FIXED(<number>, <decimals>, <no_commas>)
```

DAX

```
=FIXED([PctCost],3,1)
```

FixedWithComma	FixedWithoutComma
334,251.453	334251.453
19,428,760.264	19428760.264
43,304.553	43304.553

Text Function

- برای فرمت دهی اعداد ، تاریخ و زمان قابل استفاده می باشد خروجی این تابع رشته می باشد

```
DAX
FORMAT(<value>, <format_string>)
```

```
DAX
FORMAT( 12345.67, "General Number")
FORMAT( 12345.67, "Currency")
FORMAT( 12345.67, "Fixed")
FORMAT( 12345.67, "Standard")
FORMAT( 12345.67, "Percent")
FORMAT( 12345.67, "Scientific")
```

General Number	Currency	Standard	Percent	Scientific
12345.67	\$12,345.67	12,345.67	1234567.00%	1.23E+04

Text Function

- Left, Len, Lower, Mid:
 - تابع Left : از مقدار سمت چپ رشته بر اساس تعداد کاراکتر ورودی مقادیری را برمی گرداند.
 - تابع Len : طول یک رشته را مشخص می کند
 - تابع Lower : مقادیر رشته ای را به حروف کوچک تبدیل می کند.
 - تابع Mid : از طریق این تابع می توانیم از هر مکانی در داخل رشته مقادیری را برمی گردانیم.

Text Function

- Replace, Rept, Right, Search:
 - تابع Replace : عمل جایگزینی مقدار قبلی را با مقدار جدید از مکانی مشخص ، انجام می دهد.
 - تابع Rept : عمل تکرار را انجام می دهد مخفف Repeat می باشد
 - تابع Right : از مقدار سمت راست رشته بر اساس تعداد کاراکتر ورودی مقادیری را برمی گرداند.
 - تابع Search : بررسی دو رشته در داخل هم را انجام می دهد در صورتی که کاراکتر در رشته مورد نظر وجود نداشته باشد خطا برمی گرداند

Text Function

- Substitute:

- این تابع شبیه به تابع Replace عمل جایگزینی را انجام می دهد.
- در صورتی که بخواهید متن خاصی را جایگزین کنید از تابع فوق استفاده کنید ، در صدرتی که می خواهید در یک مکتب خاصی عمل جایگزینی را انجام دهید از Replace استفاده کند.

DAX

```
SUBSTITUTE(<text>, <old_text>, <new_text>, <instance_num>)
```

DAX

```
=SUBSTITUTE([Product Code], "NW", "PA")
```

Text Function

- Unichar:
 - این تابع از طریق مقدار عددی که برایش وارد می شود شکل نمایشی Unicode را بر روی صفحه نمایش می دهد.
-

DAX
UNICHAR(number)

128526	128544	10026	128072	128073	128070	128071
😊	😊	⌚	👉	👉	👉	👉
😊	😊	⌚	👉	👉	👉	👉

مثال ۲ - ایجاد Indicator از طریق Html Entity

Company	Rank	Stars
Donec Institute	1	★☆☆☆☆
Elit Nulla Facilisi LLP	3	★★★☆☆
Euismod Mauris Eu PC	2	★★☆☆☆
Massa Mauris Consulting	5	★★★★★
Risus Nulla Consulting	4	★★★★☆
Total	15	

توابع تاریخ و زمان Date and Time Functions

* توابع تاریخ و زمان

این توابع در DAX شبیه به توابع تاریخ و زمان در اکسل می باشند.

محاسبه فاصله دو تاریخ ، بدست آوردن اجزای تاریخ ، ایجاد تقویم ، بدست آوردن زمان جاری و ... بر عهده این توابع می باشد.

- » [DAX functions](#)
 - » [New DAX functions](#)
 - » [Date and time functions](#)
 - » [Time-intelligence functions](#)
 - » [Filter functions](#)
 - » [Information functions](#)
 - » [Logical functions](#)
 - » [Math and trig functions](#)
 - » [Other functions](#)
 - » [Parent and child functions](#)
 - » [Statistical functions](#)
 - » [Text functions](#)

Date & Time

Function	Description
CALENDAR	Returns a table with a single column named "Date" that contains a contiguous set of dates.
CALENDARAUTO	Returns a table with a single column named "Date" that contains a contiguous set of dates.
DATE	Returns the specified date in datetime format.
DATEDIFF	Returns the count of interval boundaries crossed between two dates.
DATEVALUE	Converts a date in the form of text to a date in datetime format.
DAY	Returns the day of the month, a number from 1 to 31.
EDATE	Returns the date that is the indicated number of months before or after the start date.
EOMONTH	Returns the date in datetime format of the last day of the month, before or after a specified number of months.
HOUR	Returns the hour as a number from 0 (12:00 A.M.) to 23 (11:00 P.M.).
MINUTE	Returns the minute as a number from 0 to 59, given a date and time value.
MONTH	Returns the month as a number from 1 (January) to 12 (December).
NOW	Returns the current date and time in datetime format.
QUARTER	Returns the quarter as a number from 1 to 4.
SECOND	Returns the seconds of a time value, as a number from 0 to 59.
TIME	Converts hours, minutes, and seconds given as numbers to a time in datetime format.
TIMEVALUE	Converts a time in text format to a time in datetime format.
TODAY	Returns the current date.
UTCNOW	Returns the current UTC date and time
UTCTODAY	Returns the current UTC date.
WEEKDAY	Returns a number from 1 to 7 identifying the day of the week of a date.
WEEKNUM	Returns the week number for the given date and year according to the return type value.
YEAR	Returns the year of a date as a four digit integer in the range 1900-9999.
YEARFRAC	Calculates the fraction of the year represented by the number of whole days between two dates.

Date & Time

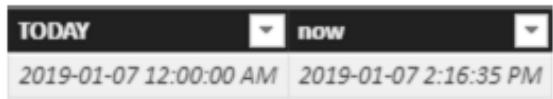
- DateDiff: اختلاف دو زمان را به حالات مختلف روز ، ماه ، سال و ... بر می گرداند

```
DAX  
DATEDIFF(<start_date>, <end_date>, <interval>)
```

```
interval:  
- SECOND  
- MINUTE  
- HOUR  
- DAY  
- WEEK  
- MONTH  
- QUARTER  
- YEAR
```

Date & Time

- Today, Now:



DAX

```
TODAY()
```

DAX

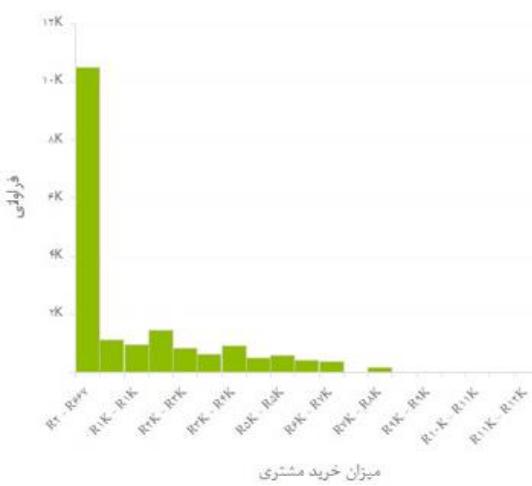
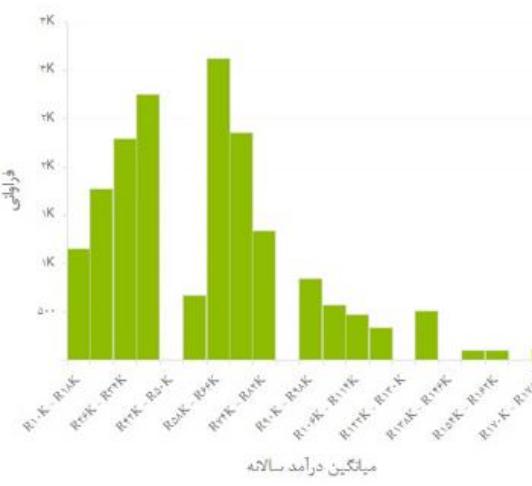
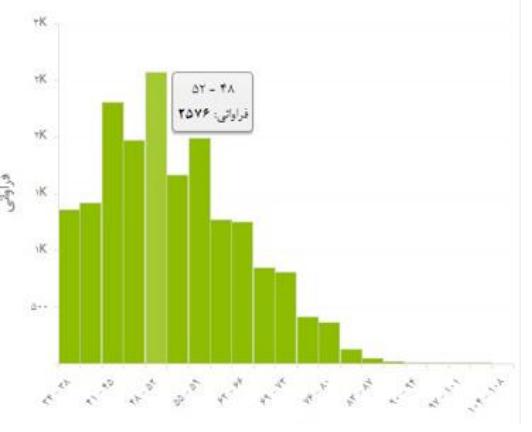
```
=YEAR(TODAY())-1963
```

DAX

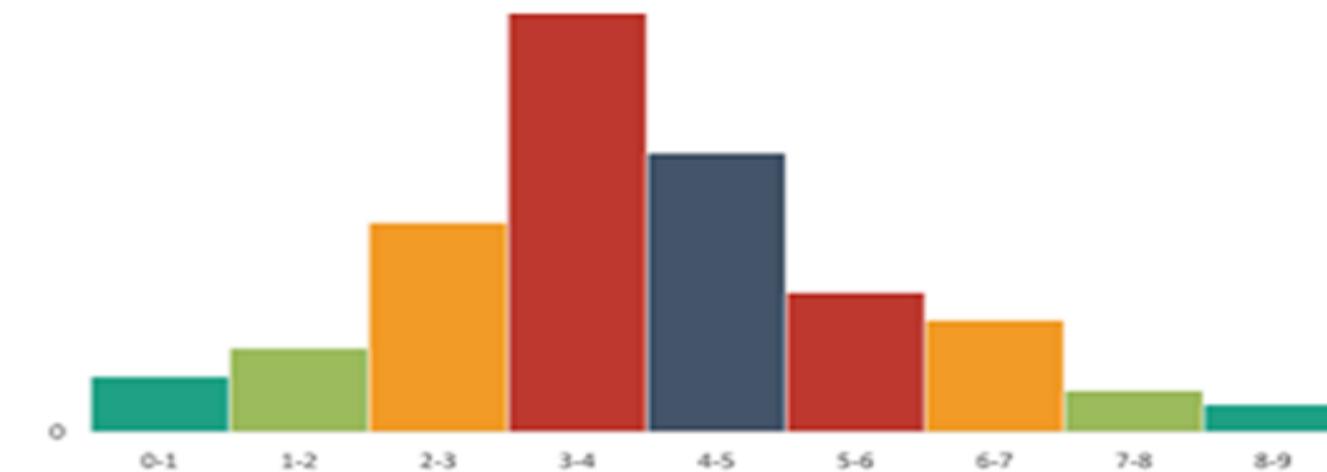
```
NOW()
```

- زمان جاری را برمی‌گرداند

مثال ۳ - نمودار هیستوگرام



- نقطه اوج **Pick** بیشترین فراوانی در کدام بازه می باشد
- تمرکز **Concentration** دو یا سه میله که اندازه های مشابه دارند
- گپ **Gap** در بازه یا بازه هایی هیچ اطلاعاتی در دسترس نباشد
- داده پرت **Outlier** بازه یا رنج هایی که خیلی بعید است داده های در آنها فراوانی داشته باشد



Time intelligence

• توابع هوشمند زمان

پیدا کردن آخرین روز از ماه ، فصل ، سال
(مثال موجودی پایان دوره یا اول دوره) ،
اضافه کردن به تاریخ ، محاسبه مقادیر با روز ،
ماه ، سال گذشته ، محاسبه داده تجمعی بر
پایه زمان و ... بر عهده این دسته از توابع می
باشد

- ✓ DAX functions
 - > New DAX functions
 - > Date and time functions
 - > Time-intelligence functions
 - > Filter functions
 - > Information functions
 - > Logical functions
 - > Math and trig functions
 - > Other functions
 - > Parent and child functions
 - > Statistical functions
 - > Text functions

Time intelligence

- Time Intelligence functions in DAX are a set of functions that give you insight from the date and time dimensions.
- These functions enable you to manipulate data using time periods, including days, months, quarters, and years, and then build and compare calculations over those periods.
- These functions need date/time dimension.

Date dimension in Time intelligence:

The date dimension that is acceptable by time intelligence functions of DAX should;

- have one record per day
- start from the minimum date in the date field or before that, and ends at the maximum date in the date field or later than that.
- have no date missing (if there are no sales in the 1st of January, still that date should be in this table. This is one of the reasons why you do need to have a separate date table)

Time intelligence

Function	Description
CLOSINGBALANCEYEAR	Evaluates the expression at the last date of the year in the current context.
DATEADD	Returns a table that contains a column of dates, shifted either forward or backward in time by the specified number of intervals from the dates in the current context.
DATESBETWEEN	Returns a table that contains a column of dates that begins with a specified start date and continues until a specified end date.
DATESINPERIOD	Returns a table that contains a column of dates that begins with a specified start date and continues for the specified number and type of date intervals.
DATESMTD	Returns a table that contains a column of the dates for the month to date, in the current context.
ENDOFMONTH	Returns the last date of the month in the current context for the specified column of dates.
FIRSTDATE	Returns the first date in the current context for the specified column of dates.
FIRSTNONBLANK	Returns the first value in the column, column, filtered by the current context, where the expression is not blank
LASTDATE	Returns the last date in the current context for the specified column of dates.
LASTNONBLANK	Returns the last value in the column, column, filtered by the current context, where the expression is not blank.
NEXTDAY	Returns a table that contains a column of all dates from the next day, based on the first date specified in the dates column in the current context.
OPENINGBALANCEMONTH	Evaluates the expression at the first date of the month in the current context.
PARALLELPERIOD	Returns a table that contains a column of dates that represents a period parallel to the dates in the specified dates column, in the current context, with the dates shifted a number of intervals either forward in time or back in time.
PREVIOUSDAY	Returns a table that contains a column of all dates representing the day that is previous to the first date in the dates column, in the current context.
SAMEPERIODLASTYEAR	Returns a table that contains a column of dates shifted one year back in time from the dates in the specified dates column, in the current context.
STARTOFMONTH	Returns the first date of the month in the current context for the specified column of dates.
TOTALMTD	Evaluates the value of the expression for the month to date, in the current context.

Time intelligence

توابع Total

محاسبه داده تجمعی ماه ، فصل ، سال

DAX

Copy

```
TOTALMTD(<expression>,<dates>[,<filter>])
```

DAX

Copy

```
TOTALQTD(<expression>,<dates>[,<filter>])
```

DAX

Copy

```
TOTALYTD(<expression>,<dates>[,<filter>][,<year_end_date>])
```

DAX

Copy

```
=TOTALQTD(SUM(InternetSales_USD[SalesAmount_USD]),DateTime[DateKey])
```

Time intelligence

CLOSINGBALANCE توابع

محاسبه مقادیر آخرین روز از ماه ، فصل ، سال توسط این توابع صورت می گیرد.
ورودی این تابع یک تاریخ مثلادی می باشد

DAX	
<code>CLOSINGBALANCEMONTH(<expression>,<dates>[,<filter>])</code>	

DAX	
<code>CLOSINGBALANCEQUARTER(<expression>,<dates>[,<filter>])</code>	

DAX	
<code>CLOSINGBALANCEYEAR(<expression>,<dates>[,<filter>][,<year_end_date>])</code>	

مثال ٤ - تابع TotalYTD

CALCULATE						
1	Sales YTD =	TOTALYTD(SUM(FactInternetSales[SalesAmount]),	FactInternetSales[OrderDate].[Date]		
2						
3						
4)					

Year	Quarter	Month	Day	SalesAmount	Sales YTD
2005	Qtr 3	July	1	14,477.34	14,477.34
2005	Qtr 3	July	2	13,931.52	28,408.86
2005	Qtr 3	July	3	15,012.18	43,421.04
2005	Qtr 3	July	4	7,156.54	50,577.58
2005	Qtr 3	July	5	15,012.18	65,589.75
2005	Qtr 3	July	6	14,313.08	79,902.83
2005	Qtr 3	July	7	7,855.64	87,758.47
2005	Qtr 3	July	8	7,855.64	95,614.11
2005	Qtr 3	July	9	20,909.78	116,523.89
2005	Qtr 3	July	10	10,556.53	127,080.42
2005	Qtr 3	July	11	14,313.08	141,393.50
2005	Qtr 3	July	12	14,134.80	155,528.30

Reference: [Click here](#)

توابع Math and Trig Functions

• توابع ریاضی و مثلثاتی

قدر مطلق ، سینوس ، لگاریتم ، حد بالا ، حد پایین ، فاکتوریل ، باقی مانده ، توان ، تابع جمع کردن (Sum,Sumx) ، جذر و ... در این دسته قرار دارند.

- » DAX functions
 - » New DAX functions
 - » Date and time functions
 - » Time-intelligence functions
 - » Filter functions
 - » Information functions
 - » Logical functions
 - » Math and trig functions
 - » Other functions
 - » Parent and child functions
 - » Statistical functions
 - » Text functions

Math & trig Functions

Function	Description
ABS	Returns the absolute value of a number.
CEILING	Rounds a number up, to the nearest integer or to the nearest multiple of significance.
ROUND	Rounds a number to the specified number of digits.
ROUNDDOWN	Rounds a number down, toward zero.
ROUNDUP	Rounds a number up, away from 0 (zero).
FLOOR	Rounds a number down, toward zero, to the nearest multiple of significance.
INT	Rounds a number down to the nearest integer.
DIVIDE	Performs division and returns alternate result or BLANK() on division by 0.
TRUNC	Truncates a number to an integer by removing the decimal, or fractional, part of the number.
QUOTIENT	Performs division and returns only the integer portion of the division result.
FACT	Returns the factorial of a number, equal to the series $1*2*3*...*$, ending in the given number.
POWER	Returns the result of a number raised to a power.
PRODUCT	Returns the product of the numbers in a column.
PRODUCTX	Returns the product of an expression evaluated for each row in a table.
SQRT	Returns the square root of a number.
SUM	Adds all the numbers in a column.
SUMX	Returns the sum of an expression evaluated for each row in a table.

Math & trig Functions

- DIVIDE:

Syntax

DAX	 Copy
<code>DIVIDE(<numerator>, <denominator> [,<alternateresult>])</code>	

Parameters

Term	Definition
numerator	The dividend or number to divide.
denominator	The divisor or number to divide by.
alternateresult	(Optional) The value returned when division by zero results in an error. When not provided, the default value is BLANK().

Math & trig Functions

- SUM:
 - جمع کردن تمام مقادیر یک ستون : Sum
 - این تابع در دسته Aggregator Function ها قرار دارد

DAX

```
SUM(<column>)
```

DAX

```
=SUM(Sales[Amt])
```

Math & trig Functions

- SUMX:
 - این تابع در دسته توابع Iterator قرار دارد.
 - این تابع عمل را سطر به سطر انجام می‌دهد.
 - می‌تواند بر روی بیش از یک فیلد اعمال شود

DAX

```
SUMX(<table>, <expression>)
```

```
Sales =  
SUMX (  
    Transactions,  
    Transactions[Quantity] * Transactions[UnitPrice])
```

Math & trig Functions

- SUMX:

```
1 Sales[SalesAmount] := Sales[Quantity] * Sales[Unit Price]
2 Sales[TotalProductCost] := Sales[Quantity] * Sales[Unit Cost]
3 Sales[GrossMargin] := Sales[SalesAmount] - Sales[TotalProductCost]
4 Sales[Gross Margin %] := DIVIDE ( SUM ( Sales[GrossMargin] ), SUM (Sales[SalesAmount] ) )
```

```
1 Sales[Gross Margin %] :=
2 DIVIDE (
3     SUMX ( Sales, Sales[Quantity] * Sales[Unit Price] )
4     - SUMX ( Sales, Sales[Quantity] * Sales[Unit Cost] ),
5     SUMX ( Sales, Sales[Quantity] * Sales[Unit Price] )
6 )
```

مثال ٥ - تابع SUM و SUMX

29.36M

Sum of Sales

\$12.08M

Total Margin

EnglishEducation	Sum of Sales	Sum of Margin	Total Margin	Percentage of Margin	Sum Of Margin for High School
Bachelors	9,900,142.76	4,064,815.22	\$12,080,883.65	33.65 %	
Graduate Degree	5,460,560.25	2,252,766.47	\$12,080,883.65	18.65 %	
High School	4,638,026.07	1,896,523.38	\$12,080,883.65	15.70 %	1,896,523.38
Partial College	7,723,542.88	3,186,803.34	\$12,080,883.65	26.38 %	
Partial High School	1,636,405.26	679,975.24	\$12,080,883.65	5.63 %	
Total	29,358,677.22	12,080,883.65	\$12,080,883.65	100.00 %	1,896,523.38

Statistical Functions

Function	Description
APPROXIMATEDIST	Returns the <i>approximate</i> number of rows that contain distinct values in a column.
AVERAGE	Returns the average (arithmetic mean) of all the numbers in a column.
AVERAGEA	Returns the average (arithmetic mean) of the values in a column.
AVERAGEX	Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.
COUNT	Counts the number of cells in a column that contain numbers.
COUNTBLANK	Counts the number of blank cells in a column.
COUNTROWS	Counts the number of rows in the specified table, or in a table defined by an expression.
DISTINCTCOUNT	Counts the number of distinct values in a column.
MAX	Returns the largest numeric value in a column, or between two scalar expressions.
MEDIAN	Returns the median of numbers in a column.
MIN	Returns the smallest numeric value in a column, or between two scalar expressions.
RANK.EQ	Returns the ranking of a number in a list of numbers.
RANKX	Returns the ranking of a number in a list of numbers for each row in the <i>table</i> argument.
STDEV.P	Returns the standard deviation of the entire population.
STDEV.S	Returns the standard deviation of a sample population.
VAR.P	Returns the variance of the entire population.

Statistical Functions

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

- AVERAGE:

Syntax

DAX	<code>AVERAGE(<column>)</code>	 Copy
-----	--------------------------------------	--

- AVERAGEA

Returns the average (arithmetic mean) of the values in a column. Handles text and non-numeric values.

Syntax

DAX	<code>AVERAGEA(<column>)</code>	 Copy
-----	---------------------------------------	--

- AVERAGEX:

Calculates the average (arithmetic mean) of a set of expressions evaluated over a table.

Syntax

DAX	<code>AVERAGEX(<table>,<expression>)</code>	 Copy
-----	---	--

Statistical Functions:

Example

The following example returns the average of non-blank cells in the referenced column, given the following table. If you used the AVERAGE function, the mean would be 21/2; with the AVERAGEA function, the result is 22/5.

Transaction ID	Amount	Result
0000123	1	Counts as 1
0000124	20	Counts as 20
0000125	n/a	Counts as 0
0000126		Counts as 0
0000126	TRUE	Counts as 1

DAX	 Copy
= AVERAGEA([Amount])	

Statistical Functions

تابع MAX و MAXX

- مشخص کردن بزرگترین مقدار

DAX

`MAX(<column>)`

 Copy

DAX

`MAX(<expression1>, <expression2>)`

 Copy

DAX

`=Max([TotalSales], [TotalPurchases])`

 Copy

DAX

`MAXX(<table>, <expression>)`

 Copy

Statistical Functions

- RANKX:

Returns the ranking of a number in a list of numbers for each row in the *table* argument.

Syntax

DAX

 Copy

```
RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])
```

- RANK.EQ:

Returns the ranking of a number in a list of numbers.

Syntax

DAX

 Copy

```
RANK.EQ(<value>, <columnName>[, <order>])
```

The following example creates a calculated column that ranks the values in SalesAmount_USD, from the *InternetSales_USD* table, against all numbers in the same column.

DAX

 Copy

```
= RANK.EQ(InternetSales_USD[SalesAmount_USD], InternetSales_USD[SalesAmount_USD])
```

Statistical Functions

تابع RANKX

- از این تابع برای ایجاد رتبه بندی استفاده می شود.
- مقدار آیتم ties می تواند Skip یا Dense باشد ، با انتخاب مقدار گپ (فاصله) بین رنج ها ایجاد نمی شود

DAX	
<code>RANKX(<table>, <expression>[, <value>[, <order>[, <ties>]]])</code>	

DAX	
<code>-RANKX(ALL(Products), SUMX(RELATEDTABLE(InternetSales), [SalesAmount]))</code>	

مثال ٦ - تابع RANKX

Structure Formatting Properties

X ✓ 1 Rank = RANKX(RankData,RankData[Count]*RankData[Price],,DESC,Dense)

Customer	Count	Price	Rank
Karim	10	8000	1
Saeed	5	5000	2
Ali	4	4000	3
Reza	1	16000	3
Mina	3	3000	4
Hamed	2	2500	5

Statistical Functions

- STDEV.S: Returns the standard deviation of a sample population.

VAR.S

Syntax

DAX	 Copy
STDEV.S(<ColumnName>)	

- STDEV.P: Returns the standard deviation of the entire population.

VAR.P

Syntax

DAX	 Copy
STDEV.P(<ColumnName>)	

- STDEV.S uses the following formula:

$$\sqrt{\sum(x - \bar{x})^2/(n-1)}$$

- STDEV.P uses the following formula:

$$\sqrt{\sum(x - \bar{x})^2/n}$$

توابع Filter Functions

* توابع فیلتر اطلاعات

فیلتر کردن اطلاعات ، جستجو در اطلاعات و ...

. این توابع به شما کمک می کند که محاسبات

پویایی را بر روی داده ها ایجاد نمایید

- ↳ DAX functions
 - ↳ New DAX functions
 - ↳ Date and time functions
 - ↳ Time-intelligence functions
 - ↳ Filter functions
 - ↳ Information functions
 - ↳ Logical functions
 - ↳ Math and trig functions
 - ↳ Other functions
 - ↳ Parent and child functions
 - ↳ Statistical functions
 - ↳ Text functions

Context:

- **Context** به شیوه اعمال لایه های مختلف فیلتر داده بر روی جداول به منظور محاسبه مقادیر مورد نظر گفته می شود.
- اکثر فیلترها بدون نیاز به نوشتن در فرمول DAX اعمال می شود.
- بصورت داینامیک مقادیر متناظر در سطر و ستون جداول محاسبه میشود.
- دو نوع وجود دارد: Row context و Filter context
- هر مقدار در گزارش بصورت جداگانه محاسبه می شود و از خروجی مقدار دیگری استفاده نمی کند.

Filter Context

SalesTerritoryGroup	CalendarYear	EnglishProductName	CategoryName	Sales Qty
Europe	2005	Bikes		231
Europe	2006	Bikes		731
Europe	2007	Accessories		4457
Europe	2007	Bikes		1983
Europe	2007	Clothing		1071
Europe	2008	Accessories		6217
Europe	2008	Bikes		1957
Europe	2008	Clothing		1442
North America	2005	Bikes		388
North America	2006	Bikes		1087

- این فیلتر بر روی یک یا چند ستون اعمال می شود.
- محاسبه هر سلول توسط یکتابع صورت می گیرد که فیلترهای متناظر با آن سلول، ورودی آن هستند.
- این فیلتر مشخص می کند که هر سطر جدول باقیستی در محاسبه بکار گرفته شود یا خیر
- شامل دو نوع ضمنی (**Implicit**) و صریح (**Explicit**) است

Implicit Filter Context

- در این نوع فیلتر تابعی استفاده نمی شود و بصورت خودکار اعمال می گردد.
- به آن **query context** هم گفته می شود.

SalesTerritoryGroup	2005	2006	2007	2008	Total
Europe	231	731	7511	9616	18089
North America	388	1087	11597	15892	28964
Pacific	394	859	5335	6757	13345
Total	1013	2677	24443	32265	60398

SELECT

```
SUM([OrderQuantity]) AS [Sales Qty]  
FROM FactInternetSales AS S  
INNER JOIN DimSalesTerritory AS C  
    ON C.[SalesTerritoryKey] =  
S.[SalesTerritoryKey] INNER JOIN DimDate AS D  
    ON D.DateKey = S.[OrderDateKey]
```

WHERE

```
C.[SalesTerritoryGroup] = 'Europe'  
AND D.[Calendar Year] = 2005
```

Implicit Filter Context

Person	Pet	Value
Andrew	Cat	10
Andrew	Dog	20
Andrew	Dog	30
Bradley	Cat	40
Bradley	Cat	50
Bradley	Dog	60
Charles	Cat	70
Charles	Bird	80
Charles	Dog	90

Table 3-2. Dataset from Table 3-1 with Filter Applied to [Person] Column

Person	Pet	Value
Andrew	Cat	10
Andrew	Dog	20
Andrew	Dog	30

Person	Bird	Cat	Dog	Total
Andrew		10	50	60
Bradley			90	60
Charles	80	70	90	240
Total	80	170	200	450

Implicit Filter Context

```
Sales Qty overridden = IF( SELECTEDVALUE(DimDate[CalendarYear])=2007,  
    -- Then --  
    "overridden",  
    --Else--  
    SUM(FactInternetSales[OrderQuantity])  
)
```

SalesTerritoryGroup	2005	2006	2007	2008	Total
Europe	231.00	731.00	overridden	9,616.00	18,089.00
NA			overridden		
North America	388.00	1,087.00	overridden	15,892.00	28,964.00
Pacific	394.00	859.00	overridden	6,757.00	13,345.00
Total	1,013.00	2,677.00	overridden	32,265.00	60,398.00

Explicit Filter Context

- این نوع فیلتر از طریق فرمول نویسی در محاسبات اعمال می شود.
- می تواند در کنار فیلترهای ضمنی و ردیفی و یا بجای آنها بکار گرفته شوند.
- با استفاده از این فیلتر مقادیری که با فیلترهای ضمنی و ردیفی قابل بدست آوردن نیست، محاسبه می شود.
- از توابعی نظیر ALLSELECTED، ALLEXCEPT، ALL، FILTER و ... در توابعی نظیر COUNTROWS و ... استفاده می شود.

مثال ۷ - فیلترهای ضمئی و صریح

SalesTerritoryGroup	Sales Qty	Filter on Europe as Column	Filter on Europe as Measure
Europe	18089	18089	18089
NA			18089
North America	28964		18089
Pacific	13345		18089
Total	60398	18089	18089

```
Filter on Europe as Column = CALCULATE(  
    SUM(FactInternetSales[OrderQuantity]),  
    DimSalesTerritory[SalesTerritoryGroup] = "Europe"  
)
```

```
Filter on Europe as Measure = CALCULATE(  
    SUM(FactInternetSales[OrderQuantity]),  
    DimSalesTerritory[SalesTerritoryGroup] = "Europe"  
)
```

مثال – ۷ HardCoded

Sales Territory Group	Sales Qty	Hardcoded Calculated Column	Hardcoded Calculated Measure
Europe	18089	18089	1
NA			1
North America	28964	28964	1
Pacific	13345	13345	1
Total	60398	60398	1

Hardcoded Calculated Measure = 1

Hardcoded Calculated Column = 1

Row Context

- *Row context* is effective when you create calculated columns or execute code inside an iterator function.
 - calculated columns perform a logical computation to generate a value for every row in the calculated column.
 - Iterators, such as *X functions*, are functions that process once for every row over any table passed to the function.

The screenshot shows a user interface for creating a calculated column. At the top, there is a text input field containing the formula: "Estimated Weight = 'Fact Sale'[Quantity] * RELATED('Dimension Stock Item'[Typical Weight Per Unit])". Below this, a preview pane displays a table with the following data:

Sale Key	Invoice Date Key	Quantity	Stock Item Key	Estimated Weight
1	Tuesday, 1 January 2013	10	153	150
2	Tuesday, 1 January 2013	9	170	1.35
3	Tuesday, 1 January 2013	9	210	0.45
4	Tuesday, 1 January 2013	3	106	1.05
5	Tuesday, 1 January 2013	96	14	9.6
6	Tuesday, 1 January 2013	5	90	1.75
7	Tuesday, 1 January 2013	2	170	0.3

Figure 3-12. Sample output of 'Fact Sale' with the new calculated column

Filter Function

Function	Description
ALL	Returns all the rows in a table, or all the values in a column, ignoring any filters that might have been applied.
ALLCROSSFILTERED	Clears all filters which are applied to a table.
ALLEXCEPT	Removes all context filters in the table except filters that have been applied to the specified columns.
ALLNOBLANKROW	From the parent table of a relationship, returns all rows but the blank row, or all distinct values of a column but the blank row, and disregards any context filters that might exist.
ALLSELECTED	Removes context filters from columns and rows in the current query, while retaining all other context filters or explicit filters.
CALCULATE	Evaluates an expression in a modified filter context.
CALCULATETABLE	Evaluates a table expression in a modified filter context.
EARLIER	Returns the current value of the specified column in an outer evaluation pass of the mentioned column.
EARLIEST	Returns the current value of the specified column in an outer evaluation pass of the specified column.
FILTER	Returns a table that represents a subset of another table or expression.
KEEPFILTERS	Modifies how filters are applied while evaluating a CALCULATE or CALCULATETABLE function.
LOOKUPVALUE	Returns the value for the row that meets all criteria specified by search conditions. The function can apply one or more search conditions.
REMOVEFILTERS	Clears filters from the specified tables or columns.
SELECTEDVALUE	Returns the value when the context for columnName has been filtered down to one distinct value only. Otherwise returns

Filter Function

- **FILTER:** این تابع یک جدول یا قسمتی از داده های یک جدول را بر می گرداند.

Syntax

DAX

Copy

```
FILTER(<table>,<filter>)
```

```
FILTER('InternetSales_USD',
        RELATED('SalesTerritory'[SalesTerritoryCountry]) <>"United States"
    )
```

Filter Function

```
Count of Sales (10 or more) =  
    COUNTROWS(  
        FILTER(  
            'Fact Sale',  
            'Fact Sale'[Quantity] > 10  
        )  
    )
```

Calendar Year	Count of Sales	Count of Sales (10 or more)
2013	60,968	30,588
2014	65,941	33,113
2015	71,898	35,550
2016	29,458	15,436
Total	228,265	114,687

Filter Function

Count of Sales (10 or more) =

```
COUNTROWS(  
    FILTER(  
        'Fact Sale',  
        'Fact Sale'[Quantity] >= 10  
        && RELATED('Dimension Stock Item'[Size]) = "M"  
    )  
)
```

Calendar Year	Count of Sales	Count of Sales (10 or more)
2013	60,968	1,534
2014	65,941	1,671
2015	71,898	1,793
2016	29,458	708
Total	228,265	5,706

مثال ۸ - تابع FILTER

Calendars	Relationships	Calculations
1 Table = FILTER (
2 FILTER (DimProduct, DimProduct[Color] = "Red"),		
3 AND (
4 CALCULATE (
5 SUM (FactInternetSales[SalesAmount]),		
6 DimDate[CalendarYear] = 2006		
7) > 100000,		
8 CALCULATE (
9 AVERAGE (FactInternetSales[SalesAmount]),		
10 DimDate[CalendarYear] = 2006		
11) > 3000		
12)		
13)		

```
SELECT *
FROM DimProduct p
WHERE
    Color = 'Red'
    AND ( SELECT SUM([SalesAmount])
        FROM [FactInternetSales] s
        INNER JOIN DimDate d
        ON s.OrderDateKey = d.DateKey
        WHERE s.ProductKey = p.ProductKey
        AND d.CalendarYear = 2006
        ) > 100000
    AND ( SELECT AVG([SalesAmount])
        FROM [FactInternetSales] s
        INNER JOIN DimDate d
        ON s.OrderDateKey = d.DateKey
        WHERE s.ProductKey = p.ProductKey
        AND d.CalendarYear = 2006
        ) > 3000
```

Filter Function

- Overriding Filters:
 - **CALCULATE**
 - **ALL**
 - **ALLEXCEPT**
 - **ALLSELECTED**

Filter Function

- **CALCULATE:** از طریق این تابع می توانیم هر عبارت محاسباتی را پردازش کنیم و همچنین فیلتری نیز بر روی آن اعمال کنیم.

Syntax

DAX	Copy
<code>CALCULATE(<expression>[, <filter1> [, <filter2> [, ...]]])</code>	

Parameters

Term	Definition
expression	The expression to be evaluated.
filter1, filter2,...	(Optional) Boolean expressions or table expressions that defines filters, or filter modifier functions.

Filter Function:

CALCULATE تابع

DAX

```
Blue Revenue =  
CALCULATE(  
    SUM(Sales[Sales Amount]),  
    'Product'[Color] = "Blue"  
)
```

Category	Sales Amount	Blue Revenue
Accessories	\$1,272,057.89	\$165,406.62
Bikes	\$94,620,526.21	\$8,374,313.88
Clothing	\$2,117,613.45	\$259,488.37
Components	\$11,799,076.66	\$803,642.10
Total	\$109,809,274.20	\$9,602,850.97

Filter Function

- **ALL:**

- در این فیلتر تمام filter context ها نادیده گرفته می شود و تمام مقادیر ستون در محاسبات در نظر گرفته می شوند.
- پارامتر اول یا جدول است و یا تعدادی ستون که همگی از یک جدول اند.
- خروجی تابع یک جدول است.

ALL = Remove_All_Implicit_Filters

Syntax

DAX

 Copy

```
ALL( [<table> | <column>[ , <column>[ , <column>[ ,...]]]] )
```

مثال ٩ – تابع ALL

CalendarYear	count of Sales	count of all sales
2005	1013	60398
2006	2677	60398
2007	5710	60398
2008	5805	60398
2009		60398
2010		60398
Total	15205	60398

- EnglishProductCategoryNa...
 - (Blank)
 - Accessories
 - Bikes
 - Clothing
 - Components

ALL:

All Reseller Sales	Column Labels				
Row Labels	Accessories	Bikes	Clothing	Components	Grand Total
2005	0.02%	9.10%	0.04%	0.75%	9.91%
2006	0.11%	24.71%	0.60%	4.48%	29.90%
2007	0.36%	31.71%	1.07%	6.79%	39.93%
2008	0.20%	16.95%	0.48%	2.63%	20.26%
Grand Total	0.70%	82.47%	2.18%	14.65%	100.00%

= SUMX(ResellerSales_USD, ResellerSales_USD[SalesAmount_USD])
/SUMX(ALL(ResellerSales_USD), ResellerSales_USD[SalesAmount_USD])

مثال ٩ - تابع ALL

CalendarYear	EnglishProductCategoryName	count of Sales	Count of all (Fact Sales) Sales	Count of all (Dim Date) Sales
2005			60398	
2005	Accessories		60398	36092
2005	Bikes	1013	60398	15205
2005	Clothing		60398	9101
2005	Components		60398	
2006			60398	
2006	Accessories		60398	36092
2006	Bikes	2677	60398	15205
2006	Clothing		60398	9101
2006	Components		60398	
2007			60398	
2007	Accessories	15025	60398	36092
2007	Bikes	5710	60398	15205
2007	Clothing	3708	60398	9101
2007	Components		60398	
2008			60398	
2008	Accessories	21067	60398	36092
2008	Bikes	5805	60398	15205
Total		60398	60398	60398

مثال ۹ - تابع ALL در Explicit Filter

Structure	Formatting	Properties
1 Count of all (Explicit) Sales =	<code>CALCULATE(COUNTROWS(FactInternetSales), FactInternetSales[SalesAmount]>10, ALL(FactInternetSales))</code>	
2006	2677	60398 38601
2007	24443	60398 38601
2008	32265	60398 38601
2009		60398 38601
2010		60398 38601
Total	60398	60398 38601

ALL = Remove_All_Implicit_Filters

Filter Function

- **ALLEXCEPT:**
 - فیلترهای مربوط به ستون در جدول مشخص شده اعمال نمی شود.
 - این تابع مشخص می کند که کدام یک از فیلترهای ضمنی فعل باقی بماند.
 - متمم تابع **ALL** است.

Syntax

DAX	 Copy
<code>ALLEXCEPT(<table>,<column>[,<column>[,...]])</code>	

Parameters

Term	Definition
table	The table over which all context filters are removed, except filters on those columns that are specified in subsequent arguments.
column	The column for which context filters must be preserved.

ALLEXCEPT:

Table 6-1. Dataset to Be Used as Table1 Table

ColumnA	ColumnB	ColumnC	ColumnD	ColumnE
A	Apple	Australia	Athletics	10
B	Banana	Brazil	Athletics	15
C	Cherry	Canada	Cricket	30

```
ALL Measure =
```

```
    CALCULATE(  
        SUM('Table1'[ColumnE]),  
        ALL(  
            Table1[ColumnA],  
            Table1[ColumnB],  
            Table1[ColumnC]  
        )  
    )
```

```
ALLEXCEPT Measure =
```

```
    CALCULATE(  
        SUM('Table1'[ColumnE]),  
        ALLEXCEPT(  
            Table1,  
            Table1[ColumnD]  
        )  
    )
```

ColumnA	ColumnB	ColumnC	ColumnD	ColumnE	ALL Measure	ALLEXCEPT Measure
A	Apple	Australia	Athletics	10	25	25
B	Banana	Brazil	Athletics	15	25	25
C	Cherry	Canada	Cricket	30	30	30
Total				55	55	55

مثال ١٠ - تابع ALLEXCEPT

EnglishProd...	EnglishProductSubcategoryName	Sales Amount	EnglishProductSubcategoryName	All Except
<input type="checkbox"/> (Blank)	Mountain Bikes	9,952,759.56	Bike Racks	39,360.00
<input type="checkbox"/> Accessories	Road Bikes	14,520,584.04	Bike Stands	39,591.00
<input checked="" type="checkbox"/> Bikes	Touring Bikes	3,844,801.05	Bottles and Cages	56,798.19
<input type="checkbox"/> Clothing	Total	28,318,144.65	Caps	19,688.10
<input type="checkbox"/> Components			Cleaners	7,218.60
.			Fenders	46,619.58
.			Gloves	35,020.70
.			Helmets	225,335.60
.			Hydration Packs	40,307.67
.			Jerseys	172,950.68
.			Mountain Bikes	9,952,759.56
.			Road Bikes	14,520,584.04
.			Shorts	71,319.81
.			Socks	5,106.32
.			Total	29,358,677.22

Filter Function

- **ALLSELECTED:**

- مانند تابع ALL فیلترهای ضمنی را نادیده می گیرد با این تفاوت که تنها فیلترهای ضمنی فعال جدول را نادیده می گیرد. به عبارت دیگر:
 - تمام سطرهای جدول یا فیلد مشخص شده در نظر گرفته می شود.
 - فیلترهای اعمال شده از سایر بخش ها اعمال می گردد.

Syntax

DAX

Copy

```
ALLSELECTED([<tableName> | <columnName>[, <columnName>[, <columnName>[,...]]]])
```

مثال ۱۱ - تابع ALL و ALLSELECTED



مثال ۱۱ - تابع ALLSELECTED

- مقدار کل فروش به تفکیک گروه و سال
- Reseller Sales Amount = $\text{SUM}(\text{FactResellerSales}[SalesAmount])$
- مقدار کل فروش
- Reseller Grand Total = $\text{CALCULATE}(\text{SUM}(\text{FactResellerSales}[SalesAmount]), \text{ALL}(\text{FactResellerSales}))$
- فروش کل بر اساس آیتم های نمایش داده شده
- Reseller Visual Total = $\text{CALCULATE}(\text{SUM}(\text{FactResellerSales}[SalesAmount]), \text{ALLSELECTED}())$
- فروش کل بر اساس آیتم های نمایش داده شده به تفکیک گروه (جمع روی همه سال ها)
- Reseller Visual Total for All of Calendar Year =
 $\text{CALCULATE}(\text{SUM}(\text{FactResellerSales}[SalesAmount]), \text{ALLSELECTED}(\text{DimDate}[CalendarYear]))$
- فروش کل بر اساس آیتم های نمایش داده شده به تفکیک سال (جمع روی همه گروه ها)
- Reseller Visual Total for All of Product Category Name =
 $\text{CALCULATE}(\text{SUM}(\text{FactResellerSales}[SalesAmount]), \text{ALLSELECTED}(\text{DimProductCategory}[EnglishProductName]))$

مثال ۱۲ – محاسبه مقدار تجمعی Running Total

CalendarYear	Sum of Sales	Running Total All	Running Total AllSELECTED	CalendarYear
2006	6,530,343.53	9,796,717.18	6,530,343.53	<input type="checkbox"/> 2005
2007	9,791,060.30	19,587,777.48	16,321,403.82	<input checked="" type="checkbox"/> 2006 <input checked="" type="checkbox"/> 2007
2008	9,770,899.74	29,358,677.22	26,092,303.56	<input checked="" type="checkbox"/> 2008
Total	26,092,303.56	29,358,677.22	26,092,303.56	<input type="checkbox"/> 2009 <input type="checkbox"/> 2010

مثال - ۱۲

Resetting Running Total

Calendar Year	Sum of Sales	Reset Running Total All
2007	9,791,060.30	9,791,060.30
1	438,865.17	438,865.17
2	489,090.34	927,955.51
3	485,574.79	1,413,530.30
4	506,399.27	1,919,929.57
5	562,772.56	2,482,702.13
6	554,799.23	3,037,501.36
7	886,668.84	3,924,170.20
8	847,413.51	4,771,583.71
9	1,010,258.13	5,781,841.84
10	1,080,449.58	6,862,291.42
11	1,196,981.11	8,059,272.53
12	1,731,787.77	9,791,060.30
2008	9,770,899.74	9,770,899.74
1	1,340,244.95	1,340,244.95
2	1,462,479.83	2,802,724.78
3	1,480,905.18	4,283,629.96
Total	19,561,960.04	9,770,899.74

Calendar...

- 2005
- 2006
- 2007
- 2008
- 2009
- 2010

Filter Function

- **SELECTEDVALUE:**

این تابع مقدار انتخابی کاربر را مشخص می کند در صورتی
که کاربر چیزی را انتخاب نکرده باشد می توان در قسمت
مقداری **AlternateResult** را مشخص نمود
به جای این تابع می توان از تابع **HASONEVALUE** نیز
استفاده کرد

Syntax

DAX

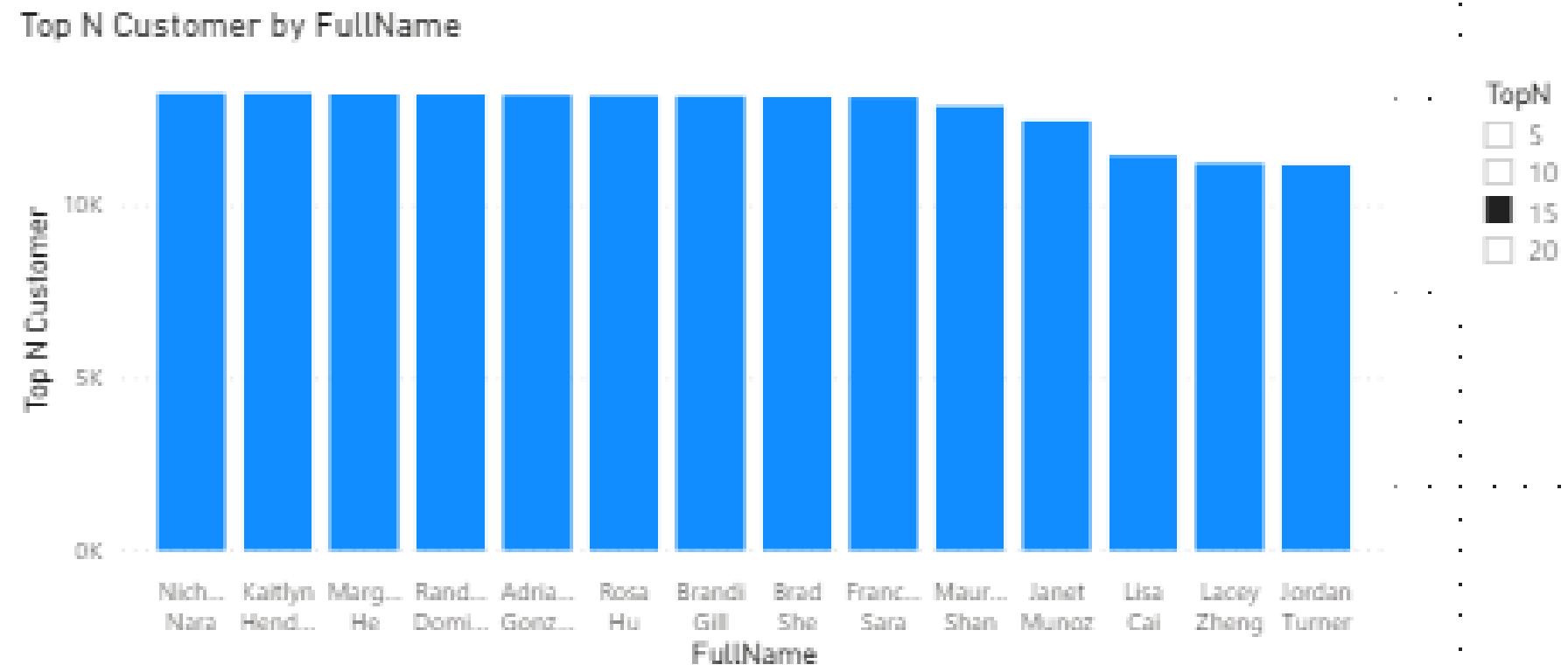
```
SELECTEDVALUE(<columnName>[, <alternateResult>])
```

Copy

مثال ۱۳ - ساخت قلم آماری داینامیک



مثال ۱۴ - نمایش داینامیک N خریدار برتر



Filter Function

- EARLIER:

به طور خاص ، این تابع عمل مقایسه ای را با رکوردهای قبلی آن انجام می دهد

Syntax

DAX

EARLIER(<column>, <number>)

مثال ١٥ – تعيين تاريخ خريد قبل

CustomerKey	OrderDateKey	Year	Month	Day
11078	20080731	2008	June	27
11287	20080731	2008	July	15
11657	20080731	2007	November	26
11794	20080731	2008	May	31
11927	20080731	2008	July	12
11981	20080731	2007	October	11
13350	20080731	2008	January	16
13671	20080731			
13753	20080731			
14029	20080731			
14114	20080731			
14474	20080731			
14680	20080731	2008	May	31
15160	20080731			
15251	20080731	2008	July	27

مثال ۱۶ - تعیین درصد تجمعی پیشرفت پروژه - تاریخ شمسی

Name	Tarikh	Percentage	Cumulative Project
آنالیز و تحلیل	1397/01/01	10	10
آنالیز و تحلیل	1397/02/02	5	15
آنالیز و تحلیل	1397/03/03	7	22
آنالیز و تحلیل	1397/04/04	8	30
آنالیز و تحلیل	1397/04/05	3	33
پیاده سازی دیتابیس	1397/04/04	12	12
پیاده سازی دیتابیس	1397/04/05	8	20
پیاده سازی دیتابیس	1397/04/06	2	22
Total		55	164

Filter Function

- **LOOKUPVALUE:**

Syntax

DAX

```
LOOKUPVALUE(  
    <result_columnName>,  
    <search_columnName>,  
    <search_value>  
    [, <search2_columnName>, <search2_value>]...  
    [, <alternateResult>]  
)
```

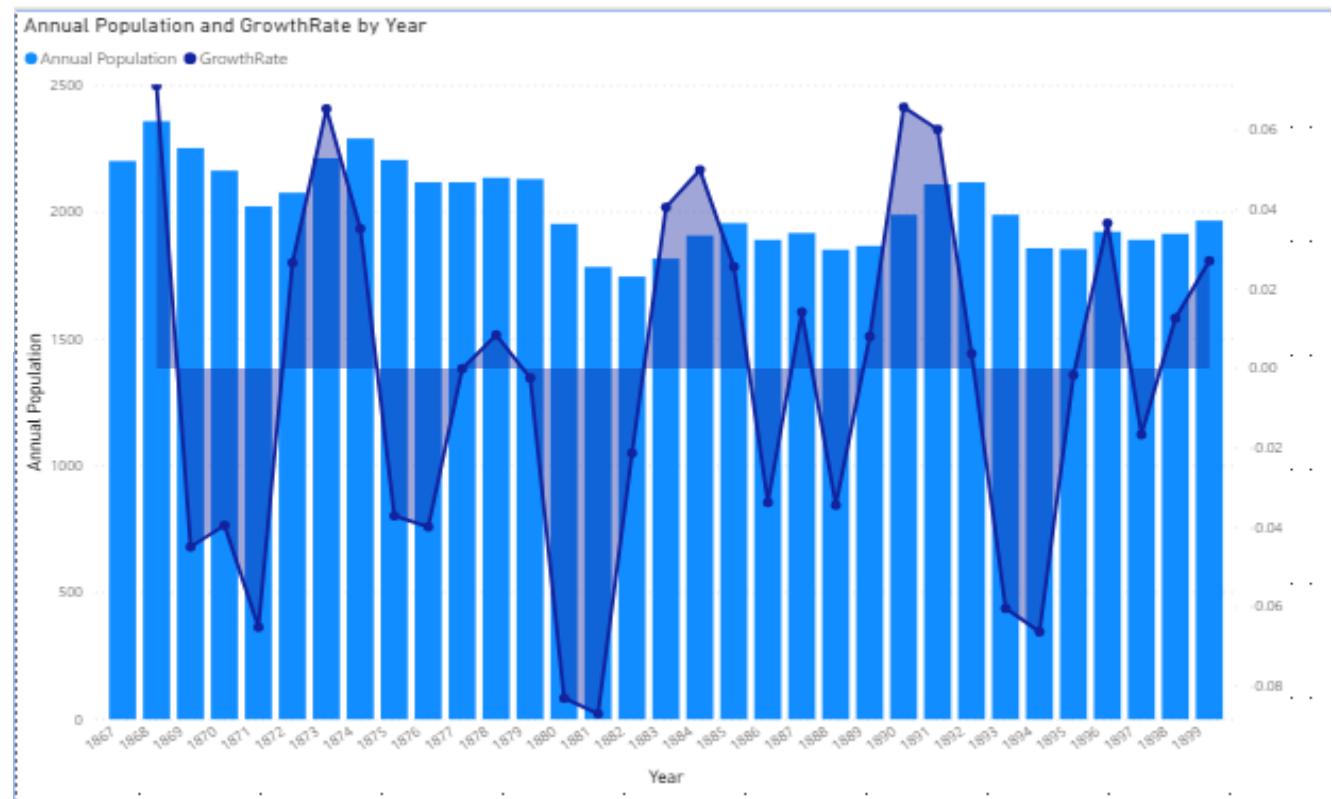
Copy

Term	Definition
result_columnName	The name of an existing column that contains the value you want to return. It cannot be an expression.
search_columnName	The name of an existing column. It can be in the same table as result_columnName or in a related table. It cannot be an expression.
search_value	A scalar expression.
alternateResult	(Optional) The value returned when the context for result_columnName has been filtered down to zero or more than one distinct value. When not provided, the function returns BLANK when result_columnName is filtered down to zero value or an error when more than one distinct value.

مثال ۱۷ - محاسبه شاخص نرخ رشد سالانه

• $(\text{مقدار سال جاری} - \text{سال گذشته}) / \text{سال گذشته}$

Year	Annual Population	Index	Population Difference
1867	2203	0	0
1868	2360	1	157
1869	2254	2	-106
1870	2165	3	-89
1871	2024	4	-141



توابع اطلاعاتی Information functions

این توابع به فیلد یا ردیف داده شده نگاه می کند و می گوید که مقدار برگردانده شده با آنچه می خواهیم مطابقت دارد یا خیر. مثلاً مقدار فیلد تهی است یا خیر

Function	Description
CONTAINS	Returns true if values for all referred columns exist, or are contained, in those columns; otherwise, the function returns false.
CONTAINSSTRING	Returns TRUE or FALSE indicating whether one string contains another string.
HASONEFILTER	Returns TRUE when the number of directly filtered values on <i>columnName</i> is one; otherwise returns FALSE.
HASONEVALUE	Returns TRUE when the context for <i>columnName</i> has been filtered down to one distinct value only. Otherwise is FALSE.
ISBLANK	Checks whether a value is blank, and returns TRUE or FALSE.
ISCROSSFILTERED	Returns TRUE when <i>columnName</i> or another column in the same or related table is being filtered.
ISERROR	Checks whether a value is an error, and returns TRUE or FALSE.
ISFILTERED	Returns TRUE when <i>columnName</i> is being filtered directly.
ISSUBTOTAL	Creates another column in a SUMMARIZE expression that returns True if the row contains subtotal values for the column given as argument, otherwise returns False.

Information Functions

- **ISFILTERED:**
Returns TRUE when *columnName* is being filtered directly. If there is no filter on the column or if the filtering happens because a different column in the same table or in a related table is being filtered then the function returns FALSE.

مثال ۱۸ – ساخت Current Selection Box

EnglishProductSubcategoryName

- (Blank)
- Bib-Shorts
- Bike Racks
- Bike Stands
- Bottles and Cages
- Bottom Brackets
- Brakes
- Caps
- Chains
- Cleaners
- Cranksets
- Derailleurs
- Fenders
- Forks
- Gloves
- Handlebars

› 3 Subcategories

All Selected

Bike Racks, Bike Stands

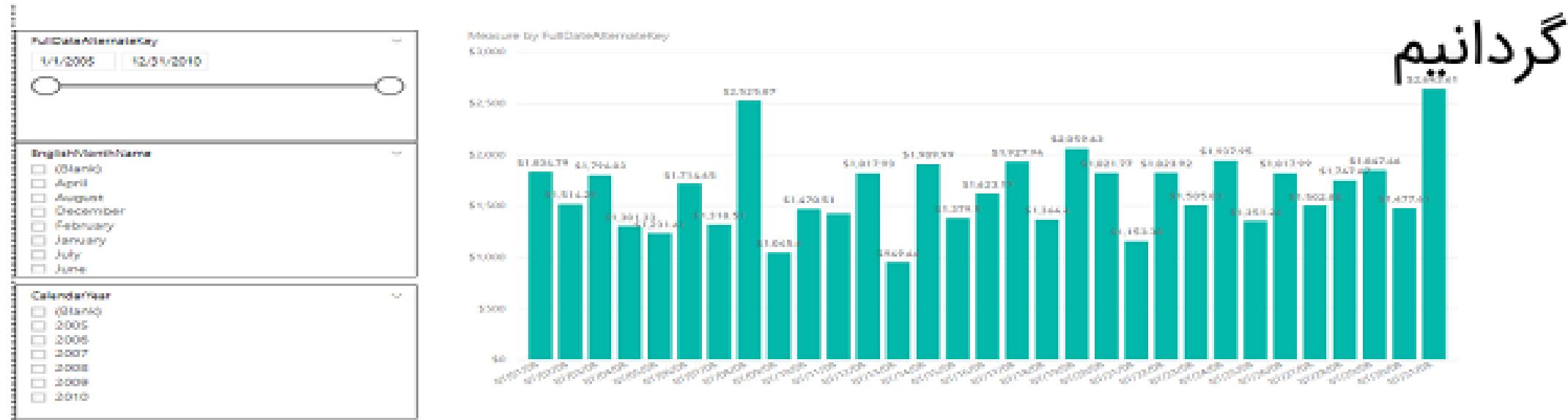
All Selected

All

All Selected

مثال ۱۹- تابع ISFILTERED() برای ایجاد Default Value

- می خواهیم در صورتی که کاربر تاریخی را فیلتر نکرده باشد به صورت پیش فرض داده ی ۳۰ روز گذشته را بر



توابع ارتباطی Relationship functions

این توابع برای مدیریت و استفاده از روابط بین جدول ها بکار می روند.

Function	Description
<u>CROSSFILTER</u>	Specifies the cross-filtering direction to be used in a calculation for a relationship that exists between two columns.
<u>RELATED</u>	Returns a related value from another table.
<u>RELATEDTABLE</u>	Evaluates a table expression in a context modified by the given filters.
<u>USERELATIONSHIP</u>	Specifies the relationship to be used in a specific calculation as the one that exists between columnName1 and columnName2.

Relationship Functions

- **USERELATIONSHIP:**

Specifies the relationship to be used in a specific calculation as the one that exists between columnName1 and columnName2.

Syntax

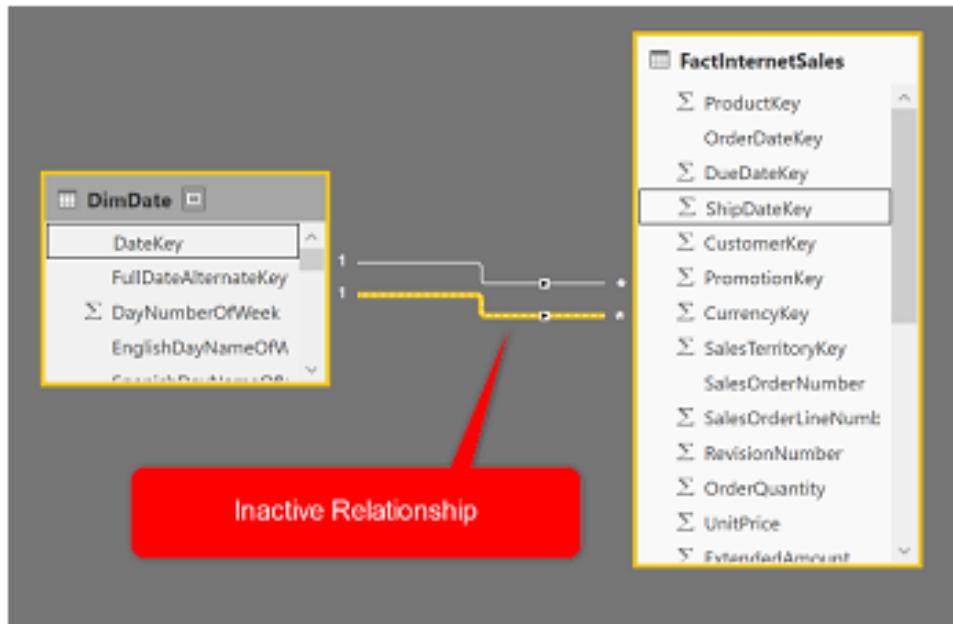
DAX

 Copy

```
USERELATIONSHIP(<columnName1>,<columnName2>)
```

ارتباط از نوع فعال و غیر فعال

- رابطه ها برای انتقال فیلتر از اهمیت بسزایی برخوردار است.
- گاهی اوقات وقتی ارتباطی بین دو جدول برقرار می شود ما بین جداول از خط تیره (Dashed Line) استفاده می شود که به این نوع ارتباط ، ارتباط از نوع Inactive گفته می شود .

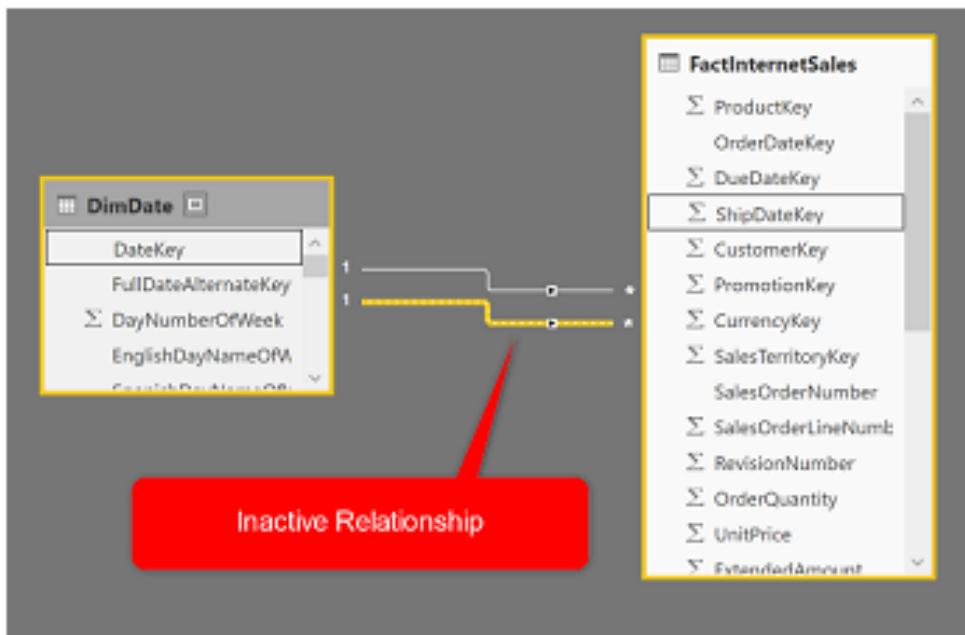


• ارتباط از نوع Inactive
ارتباط معتبری نمی باشد و هر فیلتری بر روی آن انجام شود صحیح نمی باشد .

Role Playing Dimension

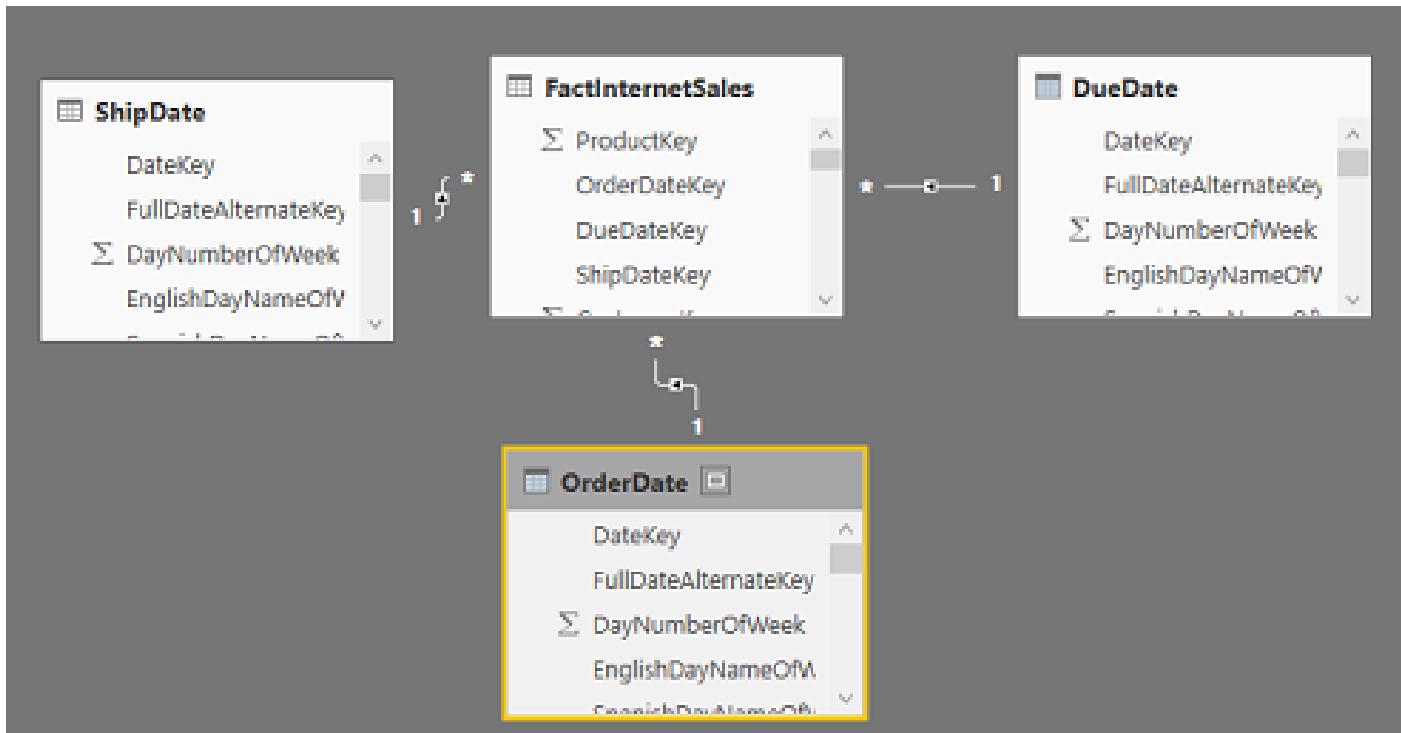
Role Playing Dimension

- بُعدی (Dimension) که نقش های متفاوتی از چندین دایمنشن را بازی می کند .
- در تصویر ذیل جدول DimDate نقش تاریخ سفارش (Order Due Date)، زمان ارسال (Ship Date) و همچنین تاریخ تحويل (Date Due) را ایفا می کند.



مثال ۲۰- پیاده سازی Dimension

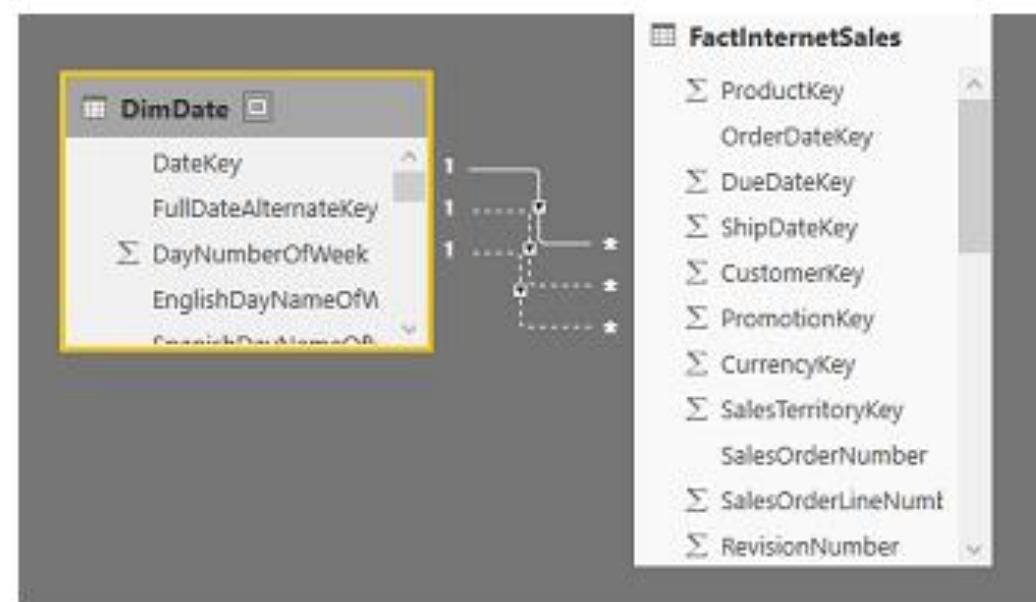
- استفاده از تابع ALL
 - از فضای RAM استفاده می کند
 - برای جداول با حجم کوچک مناسب می باشد



مثال ۲۰ - پیاده سازی Role Playing Dimension

- استفاده از تابع USERELATIONSHIP
 - در این روش فضای زیادی از RAM اشغال نمی شود.
 - باید برای هر فرمول یک قلم آماری ایجاد کرد.

```
Sales by Ship Date = CALCULATE(  
    SUM(FactInternetSales[SalesAmount]),  
    USERELATIONSHIP(  
        FactInternetSales[ShipDateKey],  
        DimDate[DateKey]  
)
```



Relationship Functions

- **CROSSFILTER:**

Specifies the cross-filtering direction to be used in a calculation for a relationship that exists between two columns.

Syntax

DAX

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

Direction

The cross-filter direction to be used. Must be one of the following:

one - Filters on the one or lookup side of the side of the relationship filter the many side.

both -

Filters on either side filter the other

none - No cross-filtering occurs along this relationship

Relationship Functions

- **CROSSFILTER:**

- در صورتی که بین دو جدول ارتباطی برقرار باشد و بخواهیم جهت ارتباط Both یا Single می توانیم از تابع Cross Filter استفاده کنیم.
- نکته : ایجاد ارتباط از نوع Both در محیط Power Pivot کارایی داشبورد (Performance) را پایین می آورد، برای برطرف کردن این مشکل می توان Model را اصلاح کرد یا از تابع CROSSFILTER استفاده کرد.

Syntax

DAX

```
CROSSFILTER(<columnName1>, <columnName2>, <direction>)
```

مثال ۲۱- پیاده سازی Cross filter

Relationship Functions

- RELATED:

برگرداندن مقادیری از ستون های مرتبط، این تابع زمانی مورد استفاده قرار می گیرد که جداول با هم ارتباط دارند و می خواهیم فیلدی از یک جدول دیگر را فیلتر کنیم .

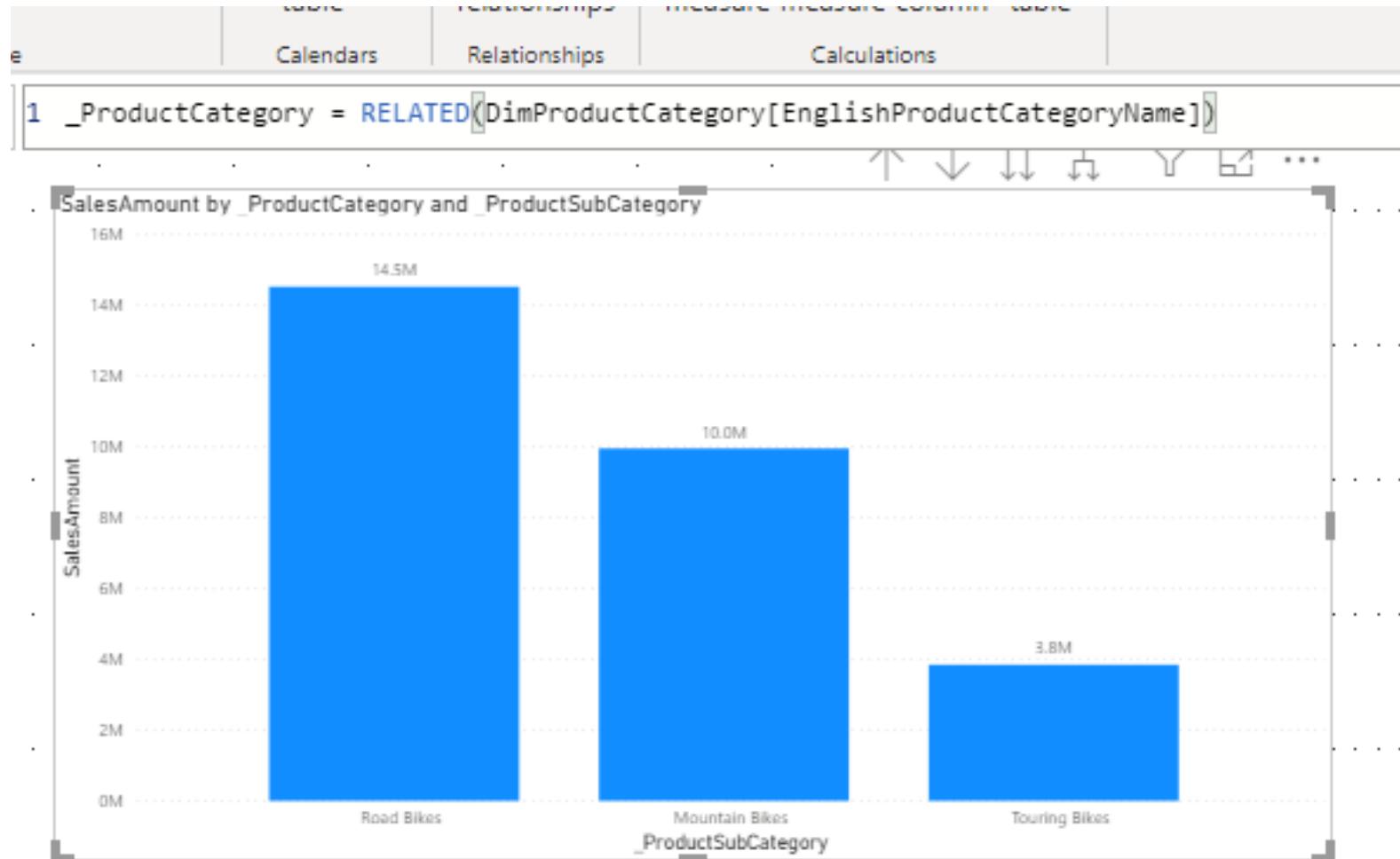
DAX

```
RELATED(<column>)
```

DAX

```
= SUMX(FILTER('InternetSales_USD'  
, RELATED('SalesTerritory'[SalesTerritoryCountry])  
<>"United States"  
)  
, 'InternetSales_USD'[SalesAmount_USD])
```

مثال ۲۲- ساخت سلسله مراتب با RELATED



Relationship Functions

- RELATEDTABLE:

خروجی این تابع یک جدول می باشد ، و همه‌ی رکوردهای که با یک رکورد مرتبط هستند را برمی گرداند . این تابع زمانی استفاده می شود که دو جدول با هم ارتباط داشته باشند

DAX

RELATEDTABLE(<tableName>)

Total Products = COUNTROWS(RELATEDTABLE(Product))

مثال ۲۳- پیاده سازی سلسله مراتب با RELATED

Structure Formatting Properties Sort Groups Relationships Calculations

X ✓ 1 Related Number =
2 | VAR RelatedNum= COUNTROWS(RELATEDTABLE(FactInternetSales))
3 | VAR Num= IF(RelatedNum=BLANK(),0,RelatedNum)
4 | Return Num

ProductCategoryKey	ProductCategoryAlternateKey	EnglishProductCategoryName	SpanishProductCategoryName	FrenchProductCategoryName	Related Number
1	1	Bikes	Bicicleta	Vélo	15205
2	2	Components	Componente	Composant	0
3	3	Clothing	Prenda	Vêtements	9101
4	4	Accessories	Accesorio	Accessoire	36092

توابع دستکاری جداول Table manipulation functions

این توابع یک جدول را ایجاد می کنند و یا جداول موجود را تغییر می دهند.

Function	Description
ADDCOLUMNS	Adds calculated columns to the given table or table expression.
CROSSJOIN	Returns a table that contains the Cartesian product of all rows from all tables in the arguments.
EXCEPT	Returns the rows of one table which do not appear in another table.
FILTERS	Returns a table of values directly applied as filters to <i>columnName</i> .
GROUPBY	Similar to the SUMMARIZE function, GROUPBY does not do an implicit CALCULATE for any extension columns that it adds.
INTERSECT	Returns the row intersection of two tables, retaining duplicates.
ROW	Returns a table with a single row containing values that result from the expressions given to each column.
SUMMARIZE	Returns a summary table for the requested totals over a set of groups.
SUMMARIZECOLUMNS	Returns a summary table over a set of groups.
TOPN	Returns the top N rows of the specified table.
UNION	Creates a union (join) table from a pair of tables.
VALUES	Returns a one-column table that contains the distinct values from the specified table or column.

تابع CROSSJOIN

- خروجی این تابع که به صورت جدول می باشد ، عمل ضرب دکارتی (Cartesian) ما بین رکوردها را انجام می دهد.

DAX	<input type="button" value="Copy"/>
CROSSJOIN(<table>, <table>[, <table>]...)	

The table Colors contains colors and patterns:

Color	Pattern
Red	Horizontal Stripe
Green	Vertical Stripe
Blue	Crosshatch

The table Stationery contains fonts and presentation:

Font	Presentation
serif	embossed
sans-serif	engraved

DAX

CROSSJOIN(Colors, Stationery)

CROSSJOIN تابع

Color	Pattern	Font	Presentation
Red	Horizontal Stripe	serif	embossed
Green	Vertical Stripe	serif	embossed
Blue	Crosshatch	serif	embossed
Red	Horizontal Stripe	sans-serif	engraved
Green	Vertical Stripe	sans-serif	engraved
Blue	Crosshatch	sans-serif	engraved

تابع ROW

- یک جدول با یک سطر (Header, Value) ایجاد می کند

DAX Copy

```
ROW(<name>, <expression>[,<name>, <expression>]...)
```

1 Table = ROW("Internet Total Sales (USD)", SUM(FactInternetSales[SalesAmount]),
2 | | "Resellers Total Sales (USD)", SUM(FactInternetSales[TaxAmt]))

Internet Total Sales (USD)	Resellers Total Sales (USD)
\$29,358,677.2207	\$2,348,694.2301

Table manipulation Functions

- **VALUES:**

- خروجی این تابع یک ستون از یک جدول می باشد که رکوردهای تکراری از داخل آن حذف شده است.
- به عنوان یک تابع مجزا نمی توانید استفاده کنید ، این تابع باید داخل توابع دیگر مثلا داخل توابع فیلتر استفاده شود.

DAX

```
VALUES(<TableNameOrColumnName>)
```

DAX

```
=COUNTRROWS(VALUES('InternetSales_USD'[SalesOrderNumber]))
```

Table manipulation Functions

- **UNION:**

Creates a union (join) table from a pair of tables.

Syntax

DAX

```
UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)
```

A table that contains all the rows from each of the two table expressions.

Table manipulation Functions

UNION(<table_expression1>, <table_expression2> [,<table_expression>]...)

- **UNION:** Remarks

- The two tables must have the same number of columns.
- Columns are combined by position in their respective tables.
- The column names in the return table will match the column names in table_expression1.
- Duplicate rows are retained.
- The returned table has lineage where possible. For example, if the first column of each table_expression has lineage to the same base column C1 in the model, the first column in the UNION result will have lineage to C1. However, if combined columns have lineage to different base columns, or if there is an extension column, the resulting column in UNION will have no lineage.
- When data types differ, the resulting data type is determined based on the rules for data type coercion.
- The returned table will not contain columns from related tables.
- This function is not supported for use in DirectQuery mode when used in calculated columns or row-level security (RLS) rules.

Table manipulation Functions

- TREATS:

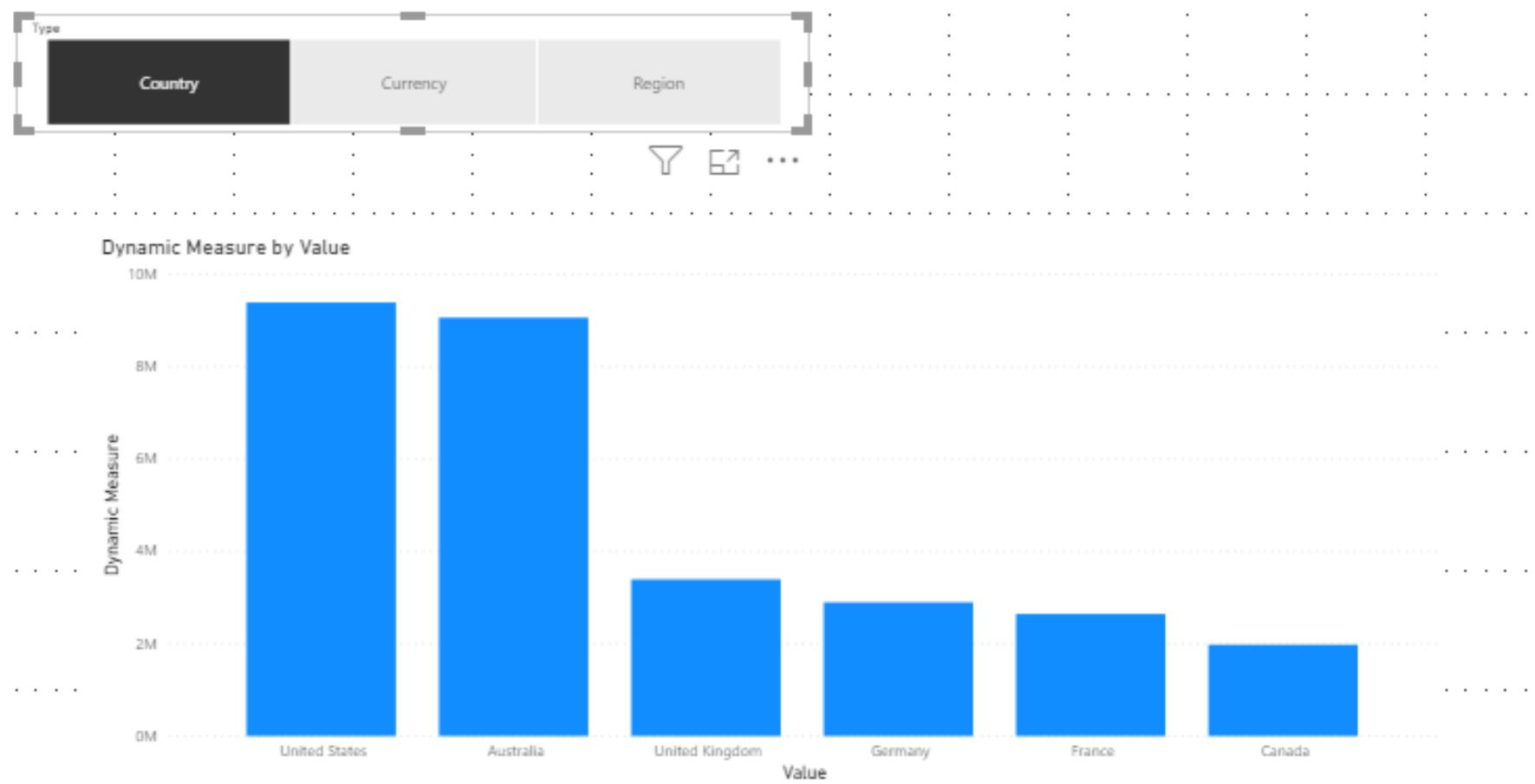
- از این تابع برای ایجاد ارتباط از نوع Virtual ما بین جداول استفاده می شود.(از این تابع زمانی استفاده می شود که ارتباط بین جداول وجود ندارد و می خواهیم به صورت منطقی ارتباط ایجاد کنیم.)



DAX

```
TREATAS(table_expression, <column>[, <column>[, <column>[,-]]]]} )
```

مثال ۲۴- انتخاب بعد داینامیک



Summarizing and Aggregating

- SUMMARIZE
- SUMMARIZECOLUMNS
- GROUPBY

Table manipulation Functions

- **SUMMARIZE:**

```
SUMMARIZE (  
    <table>,  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

- **SUMMARIZECOLUMNS:**

```
SUMMARIZECOLUMNS(  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    [<filterTable>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

- **GROUPBY:**

```
GROUPBY(  
    <table>,  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

Table manipulation Functions

- **SUMMARIZE:**

```
SUMMARIZE (  
    <table>,  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

- **SUMMARIZECOLUMNS:**

```
SUMMARIZECOLUMNS(  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    [<filterTable>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

- **GROUPBY:**

```
GROUPBY(  
    <table>,  
    <groupBy_columnName1>, [<groupBy_columnName2>,...],  
    <name1>, <expression1 >, [<name2>, <expression2 >,...],  
    )
```

Table manipulation Functions

- GROUPBY:

یک روش ساده و قدرتمند برای تجمعی اطلاعات می باشد در این تابع نیاز هست که در توابع iterator دار از تابع **CURRENTGROUP()** نیز استفاده کرد این تابع به مجموعه رکوردهای اشاره شده، اشاره می کند

DAX

```
GROUPBY (<table>, [<groupBy_columnName1>], [<name>, <expression>]... )
```

DAX

```
GROUPBY (
Sales,
Geography[Country],
Product[Category],
“Total Sales”, SUMX( CURRENTGROUP(), Sales[Price] * Sales[Qty])
)
```

Compare Running Time

Table 4-2. Time in Milliseconds of Example Using Summary Functions

DAX Function	Average Time (ms)
SUMMARIZE	40
SUMMARIZECOLUMNS	35
GROUP BY	130

تابع TOPN

- تعداد مشخصی از رکوردها را برمی‌گرداند

```
TOPN (<n_value>, <table>, <orderBy_expression>, [<order>],  
      [<orderBy_expression>, [<order>]] ...)
```

```
= SUMX (TOPN (15, Sales, Sales[Salesperson], ASC), Sales[Sales Amount])
```

EXCEPT

این تابع تمام رکوردهایی که در مجموعه اول وجود دارد و در مجموعه دوم نیست را برمی گرداند

DAX

```
EXCEPT(<table_expression1>, <table_expression2>)
```

Employees this month	Department
Alan	IT
Brittany	Sales
Caleb	Sales
Diana	HR
Evan	IT
Francisco	HR

Employees last month	Department
Alan	IT
Brittany	Sales
Diana	HR
Francisco	HR

New people = EXCEPT(Last_month;This_month)

Employees this month	Department
Evan	IT
Caleb	Sales