

Σ

مرجع جامع توابع DAX

راهنمای کامل توابع Data Analysis Expressions

فصل 1: معرفی DAX

FILTER

FILTER(<table>, <filter>)

table جدول پایه که فیلتر روی آن اعمال می‌شود
filter شرط منطقی که سطرها را فیلتر می‌کند

```
FILTER('Table', 'Table'[Item] = "Banana") =
```

SUMX

SUMX(<table>, <expression>)

table جدول حاوی سطرهایی که عبارت برای آن‌ها ارزیابی می‌شود
expression عبارت که برای هر سطر جدول ارزیابی می‌شود

```
SUMX(FILTER(InternetSales, =  
InternetSales[SalesTerritoryID]=5), [Freight])
```

MINX

MINX(<table>, <expression>)

table جدول حاوی سطرهایی که عبارت برای آن‌ها ارزیابی می‌شود
expression عبارت که برای هر سطر جدول ارزیابی می‌شود

```
MINX(FILTER('Table', 'Table'[Item] <> "Pickle"), =  
'Table'[Price])
```

SUM

SUM(<column>)

column ستونی که مقادیر آن جمع می‌شود

```
SUM('Table'[Price]) =
```

VAR

<VAR <name> = <expression

name نام متغیر
expression عبارت که مقدار متغیر را تعیین می‌کند

```
VAR __Table = FILTER('Table', 'Table'[Item] = =  
"Banana") RETURN SUMX(__Table, 'Table'[Price])
```

MAXX

MAXX(<table>, <expression>)

table جدول حاوی سطرهایی که عبارت برای آن‌ها ارزیابی می‌شود
expression عبارت که برای هر سطر جدول ارزیابی می‌شود

```
MAXX(FILTER('Table', 'Table'[Item] = "Banana"), =  
'Table'[Price])
```

COUNTROWS

COUNTROWS(<table>)

table جدولی که تعداد سطرهای آن شمارش می‌شود

```
COUNTROWS(FILTER('Table', 'Table'[Item] = =  
"Banana"))
```

فصل 2: مفاهیم اصلی بیشتر (بخش اول)

ALLSELECTED ▼

ALLSELECTED(<table> or <column>)

table or column جدول یا ستونی که فیلترهای خارجی حفظ می‌شود اما داخلی حذف می‌شود

SUMX(ALLSELECTED('Table'), 'Table'[Price]) =

ALL ☐

ALL(<table> or <column>)

table or column جدول یا ستونی که تمام فیلترها از آن حذف می‌شود

SUMX(ALL('Table'), 'Table'[Price]) =

GROUPBY ⚙️

GROUPBY(<table>, <groupBy_columnName1>, [**<name1>**, <expression1>], ...)

table جدول پایه
groupBy_columnName1 ستون برای گروه‌بندی
name1 نام ستون جدید
expression1 عبارت برای محاسبه ستون جدید (با CURRENTGROUP)

GROUPBY('Table', [Item], "Total", =
SUMX(CURRENTGROUP(), 'Table'[Price]))

SUMMARIZE =

SUMMARIZE(<table>, <groupBy_columnName1>, [**<name1>**, <expression1>], ...)

table جدول پایه
groupBy_columnName1 ستون برای گروه‌بندی
name1 نام ستون جدید
expression1 عبارت برای محاسبه ستون جدید

SUMMARIZE('Table', [Item], "Total", =
SUM('Table'[Price]))

IF ↗️

IF(<logical_test>, <value_if_true>, [**<value_if_false>**])

logical_test شرط منطقی
value_if_true مقدار اگر شرط درست
value_if_false مقدار اگر شرط غلط

IF(SUM('Table'[Price]) > 10, "High", "Low") =

AVERAGE ~

AVERAGE(<column>)

column ستونی که میانگین آن محاسبه می‌شود

AVERAGE('Table'[Price]) =

SWITCH ↔️

SWITCH(<expression>, <value1>, <result1>, [**<value2>**, <result2>], ... , [**<else>**])

expression عبارت برای ارزیابی
value1 مقدار برای مقایسه
result1 نتیجه اگر تطابق
else نتیجه پیش فرض

SWITCH(TRUE(), SUM('Table'[Price]) > 10, "High", =
"Low")

فصل 2: مفاهیم اصلی بیشتر (بخش دوم)

CONCATENATEX

```
CONCATENATEX(<table>, <expression>,  
[<delimiter>], [<orderBy_expression1>],  
[<order>])
```

table جدول برای تکرار
expression عبارت برای هر سطر
delimiter جداکننده

```
CONCATENATEX('Table', [Item], ", ") =
```

HASONEVALUE

```
HASONEVALUE(<column>)
```

column ستونی که بررسی می‌شود آیا یک مقدار منحصر به فرد دارد

```
IF(HASONEVALUE('Table'[Item]), "Single", =  
"Multiple")
```

CONTAINSROW

```
CONTAINSROW(<table>, <value1>, [<value2>],  
...)
```

table جدول برای جستجو
value1 مقدار برای تطبیق

```
IF(CONTAINSROW({ "Banana", "Apple" }, =  
'Table'[Item]), "Yes", "No")
```

ISINSCOPE

```
ISINSCOPE(<column>)
```

column ستونی که بررسی می‌شود آیا در دامنه فعلی است

```
IF(ISINSCOPE('Table'[Item]), "In Scope", "Not") =
```

UNION

```
UNION(<table1>, <table2>, ...)
```

table1 جدول اول
table2 جدول دوم برای ادغام

```
UNION('Table1', 'Table2') =
```

EXCEPT

```
EXCEPT(<table1>, <table2>)
```

table1 جدول اول
table2 جدول دوم برای کم کردن

```
EXCEPT(ALL('Table'), FILTER('Table', [Item] = =  
"Banana"))
```

فصل 3: تاریخ‌ها و تقویم‌ها (بخش اول)

YEAR 📅

YEAR(<date>)

تاریخ برای استخراج سال **date**

YEAR(TODAY()) =

DATE 📅

DATE(<year>, <month>, <day>)

سال **year**
ماه **month**
روز **day**

DATE(2024, 12, 25) =

MONTH 📅

MONTH(<date>)

تاریخ برای استخراج ماه **date**

MONTH(TODAY()) =

FORMAT ≡

FORMAT(<value>, <format_string>)

مقدار برای فرمت **value**
لگوی فرمت **format_string**

FORMAT(TODAY(), "yyyy-MM-dd") =

WEEKNUM 📅

WEEKNUM(<date>, [<return_type>])

تاریخ **date**
نوع بازگشت **return_type**

WEEKNUM(TODAY()) =

WEEKDAY 📅

WEEKDAY(<date>, [<return_type>])

تاریخ **date**
نوع بازگشت (1 برای یکشنبه) **return_type**

WEEKDAY(TODAY(), 2) =

TODAY 📅

() TODAY

بدون پارامتر تاریخ امروز را برمی‌گرداند

() TODAY =

فصل 3: تاریخ‌ها و تقویم‌ها (بخش دوم)

TOTALYTD 📄

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

expression عبارت
dates ستون تاریخ
filter فیلتر
year_end_date انتهای سال

TOTALYTD(SUM('Table'[Sales]), 'Date'[Date]) =

RANDBETWEEN 🎲

RANDBETWEEN(<low>, <high>)

low کمترین
high بیشترین

RANDBETWEEN(1, 10) =

EDATE 📅

EDATE(<start_date>, <months>)

start_date تاریخ شروع
months تعداد ماه

EDATE(TODAY(), 1) =

TOTALMTD 📅

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

expression عبارت
dates ستون تاریخ
filter فیلتر

TOTALMTD(SUM('Table'[Sales]), 'Date'[Date]) =

NETWORKDAYS 📁

NETWORKDAYS(<start_date>, <end_date>,
[<weekend>], [<holidays>])

start_date شروع
end_date پایان
weekend نوع آخر هفته
holidays تعطیلات

NETWORKDAYS(TODAY(), DATE(2024, 12, 31)) =

EOMONTH 📅

EOMONTH(<start_date>, <months>)

start_date تاریخ شروع
months تعداد ماه

EOMONTH(TODAY(), 0) =

ISODD 📄

ISODD(<number>)

number عدد برای بررسی فرد بودن

IF(ISODD(5), "Odd", "Even") =

MOD %

MOD(<number>, <divisor>)

number عدد
divisor تقسیم‌کننده

MOD(10, 3) =

فصل 4: متن (بخش اول)

LEFT ←

LEFT(<text>, <num_chars>)

text رشته
num_chars تعداد کاراکترها

LEFT("Hello", 3) =

LEN 📏

LEN(<text>)

text رشته برای محاسبه طول

LEN("Hello") =

MID ✂️

MID(<text>, <start_num>, <num_chars>)

text رشته
start_num شروع
num_chars تعداد

MID("Hello", 2, 3) =

RIGHT →

RIGHT(<text>, <num_chars>)

text رشته
num_chars تعداد کاراکترها

RIGHT("Hello", 3) =

SEARCH 🔍

SEARCH(<find_text>, <within_text>, [
<start_num>], [<not_found_value>])

find_text جستجو
within_text درون
start_num شروع
not_found_value اگر پیدا نشد (حساس به حروف نیست)

SEARCH("ELL", "Hello") =

FIND 🔍

FIND(<find_text>, <within_text>, [
<start_num>], [<not_found_value>])

find_text جستجو
within_text درون
start_num شروع
not_found_value اگر پیدا نشد

FIND("ell", "Hello") =

PATHLENGTH ~

PATHLENGTH(<path>)

path مسیر برای محاسبه طول

PATHLENGTH("1|2|3") =

SUBSTITUTE ↔

SUBSTITUTE(<text>, <old_text>, <new_text>, [
<instance_num>])

text رشته
old_text قدیم
new_text جدید
instance_num نمونه

SUBSTITUTE("Hello", "l", "x", 2) =

فصل 4: متن (بخش دوم)

UNICHAR 📄

UNICHAR(<number>)

number کد یونیکد

UNICHAR(65) =

PATHITEM 📄

PATHITEM(<path>, <position>, [<type>])

path مسیر
position مکان
type نوع (1 عدد, 2 متن)

PATHITEM("1|2|3", 2) =

hour 🕒

hour(<datetime>)

datetime زمان برای استخراج ساعت

hour(now()) =

USERPRINCIPALNAME 👤

USERPRINCIPALNAME()

بدون پارامتر نام کاربر

USERPRINCIPALNAME() =

LOOKUPVALUE 🔍

LOOKUPVALUE(<result_column>, <search_column>, <search_value>, [<search_column2>, <search_value2>], ... , [<alternate_result>])

result_column ستون نتیجه
search_column ستون جستجو
search_value مقدار جستجو
alternate_result اگر پیدا نشد

LOOKUPVALUE('Table'[Price], 'Table'[Item], "Banana")

NOW 🕒

NOW()

بدون پارامتر زمان فعلی

NOW() =

فصل 5: اعداد (بخش اول)

POWER

POWER(<number>, <power>)

number عدد
power توان

POWER(2, 3) =

AVERAGEX Σ

AVERAGEX(<table>, <expression>)

table جدول
expression عبارت

AVERAGEX(FILTER('Table', 'Table'[Item] = "Banana"), 'Table'[Price])

INT

INT(<number>)

number عدد برای بخش صحیح

INT(3.8) =

SQRT

SQRT(<number>)

number عدد برای جذر

SQRT(9) =

ROUND

ROUND(<number>, <num_digits>)

number عدد
num_digits تعداد ارقام

ROUND(3.8, 0) =

TRUNC

TRUNC(<number>, [<num_digits>])

number عدد
num_digits تعداد ارقام

TRUNC(3.8) =

EVEN

EVEN(<number>)

number عدد برای گرد به زوج

EVEN(3) =

MROUND

MROUND(<number>, <multiple>)

number عدد
multiple ضرب

MROUND(10, 3) =

فصل 5: اعداد (بخش دوم)

CURRENCY \$

CURRENCY(<value>)

value مقدار برای تبدیل به ارز

CURRENCY(123.45) =

ODD 3

ODD(<number>)

number عدد برای گرد به فرد

ODD(4) =

MEDIAN ≠

MEDIAN(<column>)

column ستون برای میانه

MEDIAN('Table'[Price]) =

DIVIDE ÷

DIVIDE(<numerator>, <denominator>, [<alternate_result>])

numerator عدد بالا

denominator عدد پایین

alternate_result اگر خطا

DIVIDE(10, 0, 0) =

RANKX ≡

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

table جدول

expression عبارت

order ترتیب (DESC/ASC)

ties نوع تساوی (Skip/Dense)

RANKX(ALL('Table'), SUM('Table'[Price]), , DESC, Dense) =

MODE ☒

MODE(<column>)

column ستون برای مود

MODE('Table'[Value]) =

LINEST ~

LINEST(<known_y>, <known_x>, [<const>], [<stats>])

known_y ستون y

known_x ستون x

const ثابت

stats آمار

LINEST('Table'[Y], 'Table'[X]) =

RANK 1 2 3

RANK(<ties>, <scope>, [<orderBy>], ...)

ties نوع تساوی (DENSE/SKIP)

scope دامنه (ROWS/GROUPS)

orderBy ترتیب

RANK(DENSE, ROWS, ORDERBY('Table'[Price], DESC)) =

فصل 6: زمان و مدت زمان و فصل 7: معیارهای مشتری

فصل 7: معیارهای مشتری

COALESCE

COALESCE(<value1>, <value2>, ...)

مقدار اول و غیره، اولین غیر خالی را برمی‌گرداند

COALESCE(BLANK(), 10) =

فصل 6: زمان و مدت زمان

TIME

TIME(<hour>, <minute>, <second>)

ساعت
دقیقه
ثانیه

TIME(12, 0, 0) =

TIMEVALUE

TIMEVALUE(<time_text>)

رشته زمان

TIMEVALUE("12:00 PM") =

DATEDIFF

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

شروع
پایان
فاصله (MINUTE, HOUR, DAY)

DATEDIFF(TODAY(), DATE(2024, 12, 31), DAY) =

ROUNDUP

ROUNDUP(<number>, <num_digits>)

عدد
ارقام

ROUNDUP(3.2, 0) =

GENERATE

GENERATE(<table1>, <table2>)

جدول اول
جدول دوم برای ضرب کرارسی

GENERATE('Table1', 'Table2') =

فصل 8: معیارهای کارمند و فصل 9: معیارهای پروژه

فصل 8: معیارهای کارمند

RADIANS

RADIANS(<degrees>)

degrees درجه

RADIANS(180) =

فصل 9: معیارهای پروژه

RELATED

RELATED(<column>)

column ستون مرتبط

RELATED('RelatedTable'[Value]) =

RELATEDTABLE

RELATEDTABLE(<table>)

table جدول مرتبط

COUNTROWS(RELATEDTABLE('RelatedTable')) =

EARLIER

EARLIER(<column>, [<num_levels>])

column ستون

num_levels سطح

FILTER('Table', 'Table'[Value] > =
EARLIER('Table'[Value]))

COUNTX

COUNTX(<table>, <expression>)

table جدول

expression عبارت

COUNTX('Table', 'Table'[Value]) =

PRODUCTX

PRODUCTX(<table>, <expression>)

table جدول

expression عبارت

PRODUCTX('Table', 'Table'[Value]) =

فصل 10: معیارهای زنجیره تأمین و فصل 11: هندسه (بخش اول)

فصل 10: معیارهای زنجیره تأمین

XIRR ↗

XIRR(<table>, <values>, <dates>, [<guess>])

جدول	table
ستون مقادیر	values
ستون تاریخ	dates
تخمین	guess

XIRR('Table', [Value], [Date]) =

DATATABLE 📄

DATATABLE(<column1>, <type1>, <column2>, <type2>, ... , { { <value1>, <value2>, ... } , ... })

نام ستون	column1
نوع و غیره	type1
مقادیرها	value1

DATATABLE("Name", STRING, { { "John" } }) =

DAY 📅

DAY(<date>)

date تاریخ برای روز

DAY(TODAY()) =

فصل 11: هندسه (بخش اول)

SIN ≡

SIN(<number>)

number عدد (رادیان)

SIN(PI()) =

COS ↗

COS(<number>)

number عدد (رادیان)

COS(0) =

ACOS △

ACOS(<number>)

number عدد

ACOS(0) =

PI ○

()PI

بدون پارامتر پی

()PI =

DEGREES ↻

DEGREES(<radians>)

radians رادیان

DEGREES(PI()) =

فصل 11: هندسه (بخش دوم) و فصل 13: سناریوهای پیشرفته

فصل 13: سناریوهای پیشرفته

NOT

NOT(<logical>)

logical منطقی برای نه

NOT(TRUE()) =

فصل 11: هندسه (بخش دوم)

TAN

TAN(<number>)

number عدد (رادیان)

TAN(0) =

ATAN2

ATAN2(<y>, <x>)

y	y
x	x

ATAN2(1, 1) =

SIGN

SIGN(<number>)

number عدد برای علامت

SIGN(-5) =

ABS

ABS(<number>)

number عدد برای مطلق

ABS(-5) =

فصل 14: بهینه‌سازی عملکرد و فصل 16: AI، اشکال‌زدایی و CALCULATE

فصل 16: AI، اشکال‌زدایی و CALCULATE

ERROR

ERROR(<message>)

message پیام خطا

ERROR("Error") =

ISERROR

ISERROR(<value>)

value مقدار برای بررسی خطا

ISERROR(1/0) =

IFERROR

IFERROR(<value>, <value_if_error>)

value مقدار
value_if_error اگر خطا

IFERROR(1/0, 0) =

EVALUATEANDLOG

EVALUATEANDLOG(<expression>, [<output>], [<tag>])

expression عبارت
output نوع خروجی
tag تگ

EVALUATEANDLOG(SUM('Table'[Price])) =

KEEPFILTERS

KEEPFILTERS(<expression>)

expression عبارت برای حفظ فیلترها

CALCULATE(SUM('Table'[Price]), =
KEEPFILTERS('Table'[Item] = "Banana"))

فصل 14: بهینه‌سازی عملکرد

TREATAS

TREATAS(<table_expression>, <column1>, [<column2>], ...)

table_expression جدول
column1 ستون

TREATAS(VALUE('Table1'[Column]), =
'Table2'[Column])

ISEMPTY

ISEMPTY(<table>)

table جدول برای خالی بودن

ISEMPTY(FILTER('Table', FALSE())) =