

# beta\_cdf()

Article • 03/09/2023

Returns the standard cumulative beta distribution function.

If  $probability = \text{beta\_cdf}(x, \dots)$ , then  $\text{beta\_inv}(probability, \dots) = x$ .

The beta distribution is commonly used to study variation in the percentage of something across samples, such as the fraction of the day people spend watching television.

## Syntax

```
beta_cdf(x, alpha, beta)
```

## Parameters

Name	Type	Required	Description
x	int, long, or real	✓	A value at which to evaluate the function.
alpha	int, long, or real	✓	A parameter of the distribution.
beta	int, long, or real	✓	A parameter of the distribution.

## Returns

The cumulative beta distribution function  $\beta$ .

### ⓘ Note

- If any argument is nonnumeric, the function returns `null`.
- If  $x < 0$  or  $x > 1$ , the function returns `NaN`.
- If  $\text{alpha} \leq 0$  or  $\text{alpha} > 10000$ , the function returns `NaN`.
- If  $\text{beta} \leq 0$  or  $\text{beta} > 10000$ , the function returns `NaN`.

## Examples

## Run the query

Kusto

```
datatable(x:double, alpha:double, beta:double, comment:string)
[
    0.9, 10.0, 20.0, "Valid input",
    1.5, 10.0, 20.0, "x > 1, yields NaN",
    double(-10), 10.0, 20.0, "x < 0, yields NaN",
    0.1, double(-1.0), 20.0, "alpha is < 0, yields NaN"
]
| extend b = beta_cdf(x, alpha, beta)
```

## Output

x	alpha	beta	comment	b
0.9	10	20	Valid input	0.999999999999959
1.5	10	20	x > 1, yields NaN	NaN
-10	10	20	x < 0, yields NaN	NaN
0.1	-1	20	alpha is < 0, yields NaN	NaN

## See also

- For computing the inverse of the beta cumulative probability density function, see `beta-inv()`.
- For computing probability density function, see `beta-pdf()`.

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# beta\_inv()

Article • 03/09/2023

Returns the inverse of the beta cumulative probability density function.

If  $probability = \text{beta\_cdf}(x, \dots)$ , then  $\text{beta\_inv}(probability, \dots) = x$ .

The beta distribution can be used in project planning to model probable completion times given an expected completion time and variability.

## Syntax

```
beta_inv(probability, alpha, beta)
```

## Parameters

Name	Type	Required	Description
<i>probability</i>	int, long, or real	✓	A probability associated with the beta distribution.
<i>alpha</i>	int, long, or real	✓	A parameter of the distribution.
<i>beta</i>	int, long, or real	✓	A parameter of the distribution.

## Returns

The inverse of the beta cumulative probability density function beta\_cdf()

### ⓘ Note

- If any argument is nonnumeric, the function returns `null`.
- If  $\alpha \leq 0$  or  $\beta \leq 0$ , the function returns `null`.
- If  $probability \leq 0$  or  $probability > 1$ , the function returns `NaN`.
- Given a value for  $probability$ , `beta_inv()` seeks that value  $x$  such that  $\text{beta\_cdf}(x, \alpha, \beta) = probability$ .

## Examples

## Run the query

Kusto

```
datatable(p:double, alpha:double, beta:double, comment:string)
[
    0.1, 10.0, 20.0, "Valid input",
    1.5, 10.0, 20.0, "p > 1, yields null",
    0.1, double(-1.0), 20.0, "alpha is < 0, yields NaN"
]
| extend b = beta_inv(p, alpha, beta)
```

## Output

p	alpha	beta	comment	b
0.1	10	20	Valid input	0.226415022388749
1.5	10	20	p > 1, yields null	
0.1	-1	20	alpha is < 0, yields NaN	NaN

## See also

- For computing cumulative beta distribution function, see `beta-cdf()`.
- For computing probability beta density function, see `beta-pdf()`.

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# beta\_pdf()

Article • 05/25/2023

Returns the probability density beta function.

The beta distribution is commonly used to study variation in the percentage of something across samples, such as the fraction of the day people spend watching television.

## Syntax

```
beta_pdf(x, alpha, beta)
```

## Parameters

Name	Type	Required	Description
x	int, long, or real	✓	A value at which to evaluate the function.
alpha	int, long, or real	✓	A parameter of the distribution.
beta	int, long, or real	✓	A parameter of the distribution.

## Returns

The probability beta density function ↗.

### ⓘ Note

- If any argument is nonnumeric, the function returns `null`.
- If  $x \leq 0$  or  $1 \leq x$ , the function returns `Nan`.
- If  $\text{alpha} \leq 0$  or  $\text{beta} \leq 0$ , the function returns `Nan`.

## Examples

Run the query

```

datatable(x:double, alpha:double, beta:double, comment:string)
[
    0.5, 10.0, 20.0, "Valid input",
    1.5, 10.0, 20.0, "x > 1, yields NaN",
    double(-10), 10.0, 20.0, "x < 0, yields NaN",
    0.1, double(-1.0), 20.0, "alpha is < 0, yields NaN"
]
| extend r = beta_pdf(x, alpha, beta)

```

## Output

x	alpha	beta	comment	r
0.5	10	20	Valid input	0.746176019310951
1.5	10	20	x > 1, yields NaN	NaN
-10	10	20	x < 0, yields NaN	NaN
0.1	-1	20	alpha is < 0, yields NaN	NaN

## See also

- For computing the inverse of the beta cumulative probability density function, see [beta-inv\(\)](#).
- For the standard cumulative beta distribution function, see [beta-cdf\(\)](#).

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# bin()

Article • 03/30/2023

Rounds values down to an integer multiple of a given bin size.

Used frequently in combination with summarize by .... If you have a scattered set of values, they'll be grouped into a smaller set of specific values.

The `bin()` and `floor()` functions are equivalent

## Syntax

```
bin(value, roundTo)
```

## Parameters

Name	Type	Required	Description
<code>value</code>	int, long, real, timespan, or datetime	✓	The value to round down.
<code>roundTo</code>	int, long, real, or timespan	✓	The "bin size" that divides <code>value</code> .

## Returns

The nearest multiple of `roundTo` below `value`. Null values, a null bin size, or a negative bin size will result in null.

## Examples

### Numeric bin

Run the query

Kusto

```
print bin(4.5, 1)
```

### Output

```
print_0
```

```
4
```

## Timespan bin

[Run the query](#)

```
Kusto
```

```
print bin(time(16d), 7d)
```

### Output

```
print_0
```

```
14:00:00:00
```

## Datetime bin

[Run the query](#)

```
Kusto
```

```
print bin(datetime(1970-05-11 13:45:07), 1d)
```

### Output

```
print_0
```

```
1970-05-11T00:00:00Z
```

## Pad a table with null bins

When there are rows for bins with no corresponding row in the table, we recommend to pad the table with those bins. The following query looks at strong wind storm events in California for a week in April. However, there are no events on some of the days.

[Run the query](#)

```
Kusto
```

```

let Start = datetime('2007-04-07');
let End = Start + 7d;
StormEvents
| where StartTime between (Start .. End)
| where State == "CALIFORNIA" and EventType == "Strong Wind"
| summarize PropertyDamage=sum(DamageProperty) by bin(StartTime, 1d)

```

## Output

StartTime	PropertyDamage
2007-04-08T00:00:00Z	3000
2007-04-11T00:00:00Z	1000
2007-04-12T00:00:00Z	105000

In order to represent the full week, the following query pads the result table with null values for the missing days. Here's a step-by-step explanation of the process:

1. Use the `union` operator to add more rows to the table.
2. The `range` operator produces a table that has a single row and column.
3. The `mv-expand` operator over the `range` function creates as many rows as there are bins between `StartTime` and `EndTime`.
4. Use a `PropertyDamage` of `0`.
5. The `summarize` operator groups together bins from the original table to the table produced by the `union` expression. This process ensures that the output has one row per bin whose value is either zero or the original count.

## Run the query

Kusto

```

let Start = datetime('2007-04-07');
let End = Start + 7d;
StormEvents
| where StartTime between (Start .. End)
| where State == "CALIFORNIA" and EventType == "Strong Wind"
| union (
    range x from 1 to 1 step 1
    | mv-expand StartTime=range(Start, End, 1d) to typeof(datetime)
    | extend PropertyDamage=0
)
| summarize PropertyDamage=sum(DamageProperty) by bin(StartTime, 1d)

```

## Output

StartTime	PropertyDamage
2007-04-07T00:00:00Z	0
2007-04-08T00:00:00Z	3000
2007-04-09T00:00:00Z	0
2007-04-10T00:00:00Z	0
2007-04-11T00:00:00Z	1000
2007-04-12T00:00:00Z	105000
2007-04-13T00:00:00Z	0
2007-04-14T00:00:00Z	0

---

## Feedback

Was this page helpful?  Yes  No

Provide product feedback  | Get help at Microsoft Q&A

# bin\_at()

Article • 12/28/2022

Rounds values down to a fixed-size bin, with control over the bin's starting point.

## Syntax

```
bin_at (value, bin_size, fixed_point)
```

## Parameters

Name	Type	Required	Description
<code>value</code>	int, long, real, timespan, or datetime	✓	The value to round.
<code>bin_size</code>	int, long, real, or timespan	✓	The size of each bin.
<code>fixed_point</code>	int, long, real, timespan, or datetime	✓	A constant of the same type as <code>value</code> indicating one value of <code>value</code> , which is a <i>fixed point</i> for which <code>bin_at(fixed_point, bin_size, fixed_point) ==</code> <code>fixed_point</code> .

### ⓘ Note

If `value` is a timespan or datetime, then the `bin_size` must be a timespan.

## Returns

The nearest multiple of `bin_size` below `value`, shifted so that `fixed_point` will be translated into itself.

## Examples

Expression	Result	Comments

Expression	Result	Comments
<code>bin_at(6.5, 2.5, 7)</code>	4.5	
<code>bin_at(time(1h), 1d, 12h)</code>	-12h	
<code>bin_at(datetime(2017-05-15 10:20:00.0), 1d, datetime(1970-01-01 12:00:00.0))</code>	datetime(2017-05-14 12:00:00.0)	All bins will be at noon
<code>bin_at(datetime(2017-05-17 10:20:00.0), 7d, datetime(2017-06-04 00:00:00.0))</code>	datetime(2017-05-14 00:00:00.0)	All bins will be on Sundays

In the following example, notice that the "fixed point" arg is returned as one of the bins and the other bins are aligned to it based on the `bin_size`. Also note that each datetime bin represents the starting time of that bin:

[Run the query](#)

Kusto

```
datatable(Date:datetime, Num:int)[
    datetime(2018-02-24T15:14),3,
    datetime(2018-02-23T16:14),4,
    datetime(2018-02-26T15:14),5]
| summarize sum(Num) by bin_at(Date, 1d, datetime(2018-02-24 15:14:00.0000000))
```

## Output

Date	sum_Num
2018-02-23 15:14:00.0000000	4
2018-02-24 15:14:00.0000000	3
2018-02-26 15:14:00.0000000	5

## See also

- [bin\(\)](#)

---

## Feedback

Was this page helpful?  Yes  No

Provide product feedback  | Get help at Microsoft Q&A

# bin\_auto()

Article • 12/28/2022

Rounds values down to a fixed-size bin, with control over the bin size and starting point provided by a query property.

## Syntax

```
bin_auto (value)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	int, long, real, timespan, or datetime	✓	The value to round into bins.

To control the bin size and starting point, set the following parameters before using the function.

Name	Type	Required	Description
<i>query_bin_auto_size</i>	int, long, real, or timespan	✓	Indicates the size of each bin.
<i>query_bin_auto_at</i>	int, long, real, or timespan		Indicates one value of <i>value</i> which is a "fixed point" for which <code>bin_auto(fixed_point) == fixed_point</code> . Default is 0.

## Returns

The nearest multiple of `query_bin_auto_size` below *value*, shifted so that `query_bin_auto_at` will be translated into itself.

## Examples

Kusto

```
set query_bin_auto_size=1h;
set query_bin_auto_at=datetime(2017-01-01 00:05);
range Timestamp from datetime(2017-01-01 00:05) to datetime(2017-01-01
```

```
02:00) step 1m  
| summarize count() by bin_auto(Timestamp)
```

## Output

Timestamp	count_
2017-01-01 00:05:00.0000000	60
2017-01-01 01:05:00.0000000	56

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# binary\_and()

Article • 12/05/2022

Returns a result of the bitwise AND operation between two values.

## Syntax

```
binary_and( value1 , value2 )
```

## Parameters

Name	Type	Required	Description
value1	long	✓	The left-hand value of the bitwise AND operation.
value2	long	✓	The right-hand value of the bitwise AND operation.

## Returns

Returns logical AND operation on a pair of numbers: value1 & value2.

---

## Feedback

Was this page helpful?



Yes



No

Provide product feedback | Get help at Microsoft Q&A

# binary\_not()

Article • 12/28/2022

Returns a bitwise negation of the input value.

## Syntax

```
binary_not(value)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	long	✓	The value to negate.

## Returns

Returns logical NOT operation on a number: value.

## Example

Run the query

Kusto

```
binary_not(100)
```

## Output

**result**

-101

## Feedback

Was this page helpful?

 Yes

 No



# binary\_or()

Article • 12/05/2022

Returns a result of the bitwise `or` operation of the two values.

## Syntax

```
binary_or(value1, value2)
```

## Parameters

Name	Type	Required	Description
<i>value1</i>	long	✓	The left-hand value of the bitwise <code>OR</code> operation.
<i>value2</i>	long	✓	The right-hand value of the bitwise <code>OR</code> operation.

## Returns

Returns logical OR operation on a pair of numbers: *value1* | *value2*.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# binary\_shift\_left()

Article • 12/28/2022

Returns binary shift left operation on a pair of numbers.

## Syntax

```
binary_shift_left(value, shift)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	int	✓	The value to shift left.
<i>shift</i>	int	✓	The number of bits to shift left.

## Returns

Returns binary shift left operation on a pair of numbers:  $\text{value} \ll (\text{shift} \% 64)$ . If n is negative, a NULL value is returned.

## Example

Run the query

Kusto

```
binary_shift_left(1,2)
```

## Output

### Result

4

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# binary\_shift\_right()

Article • 12/28/2022

Returns binary shift right operation on a pair of numbers.

## Syntax

```
binary_shift_right(value, shift)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	int	✓	The value to shift right.
<i>shift</i>	int	✓	The number of bits to shift right.

## Returns

Returns binary shift right operation on a pair of numbers: `value >> (shift%64)`. If n is negative, a NULL value is returned.

## Examples

Run the query

Kusto

```
binary_shift_right(1,2)
```

## Output

Result

```
0
```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# binary\_xor()

Article • 12/28/2022

Returns a result of the bitwise xor operation of the two values.

## Syntax

```
binary_xor(value1, value2)
```

## Parameters

Name	Type	Required	Description
value1	int	✓	The left-side value of the XOR operation.
value2	int	✓	The right-side value of the XOR operation.

## Returns

Returns logical XOR operation on a pair of numbers: value1 ^ value2.

## Examples

[Run the query](#)

```
Kusto
```

```
binary_xor(1,1)
```

## Output

**Result**

```
0
```

[Run the query](#)

```
Kusto
```

```
binary_xor(1,2)
```

## Output

### Result

3

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# bitset\_count\_ones()

Article • 12/28/2022

Returns the number of set bits in the binary representation of a number.

## Syntax

```
bitset_count_ones(value)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	int	✓	The value for which to calculate the number of set bits.

## Returns

Returns the number of set bits in the binary representation of a number.

## Example

Run the query

Kusto

```
// 42 = 32+8+2 : b'00101010' == 3 bits set
print ones = bitset_count_ones(42)
```

## Output

<b>ones</b>
3

## Feedback

Was this page helpful?

 Yes

 No



# case()

Article • 12/28/2022

Evaluates a list of predicates and returns the first result expression whose predicate is satisfied.

If none of the predicates return `true`, the result of the `else` expression is returned. All `predicate` arguments must be expressions that evaluate to a `boolean` value. All `then` arguments and the `else` argument must be of the same type.

## Syntax

```
case(predicate_1, then_1, [predicate_2, then_2, ...] else)
```

## Parameters

Name	Type	Required	Description
<i>predicate</i>	string	✓	An expression that evaluates to a <code>boolean</code> value.
<i>then</i>	string	✓	An expression that gets evaluated and its value is returned from the function if <i>predicate</i> is the first predicate that evaluates to <code>true</code> .
<i>else</i>	string	✓	An expression that gets evaluated and its value is returned from the function if neither of the <i>predicate_i</i> evaluate to <code>true</code> .

## Returns

The value of the first *then\_i* whose *predicate\_i* evaluates to `true`, or the value of *else* if neither of the predicates are satisfied.

## Example

Run the query

Kusto

```
range Size from 1 to 15 step 2
| extend bucket = case(Size <= 3, "Small",
```

```
Size <= 10, "Medium",
"Large")
```

## Output

Size	bucket
1	Small
3	Small
5	Medium
7	Medium
9	Medium
11	Large
13	Large
15	Large

---

## Feedback

Was this page helpful?

Provide product feedback  | Get help at Microsoft Q&A

# ceiling()

Article • 12/28/2022

Calculates the smallest integer greater than, or equal to, the specified numeric expression.

## Syntax

```
ceiling(number)
```

## Parameters

Name	Type	Required	Description
<i>number</i>	int, long, or real	✓	The value to round up.

## Returns

The smallest integer greater than, or equal to, the specified numeric expression.

## Examples

Run the query

Kusto

```
print c1 = ceiling(-1.1), c2 = ceiling(0), c3 = ceiling(0.9)
```

## Output

c1	c2	c3
-1	0	1

## Feedback

Was this page helpful?

Yes

No

[Provide product feedback](#) | [Get help at Microsoft Q&A](#)

# coalesce()

Article • 12/28/2022

Evaluates a list of expressions and returns the first non-null (or non-empty for string) expression.

## Syntax

```
coalesce(arg, arg_2, [arg_3, ...])
```

## Parameters

Name	Type	Required	Description
arg	scalar	✓	The expression to be evaluated.

### ⓘ Note

- All arguments must be of the same type.
- Maximum of 64 arguments is supported.

## Returns

The value of the first *arg* whose value isn't null (or not-empty for string expressions).

## Example

Run the query

Kusto

```
print result=coalesce(tolong("not a number"), tolong("42"), 33)
```

## Output

result
42

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# column\_ifexists()

Article • 12/14/2022

Takes a column name as a string and a default value. Returns a reference to the column if it exists, otherwise - returns the default value.

**Deprecated aliases:** columnifexists()

## Syntax

```
column_ifexists(columnName, defaultValue)
```

## Parameters

Name	Type	Required	Description
<i>columnName</i>	string	✓	The name of the column to check if exists.
<i>defaultValue</i>	scalar	✓	The value to use if the column doesn't exist. This value can be any scalar expression. For example, a reference to another column.

## Returns

If *columnName* exists, then the column it refers to. Otherwise *defaultValue*.

## Example

Run the query

Kusto

```
// There's no column "Capital" in "StormEvents", therefore, the State column  
// will be used instead  
StormEvents | project column_ifexists("Capital", State)
```

---

## Feedback



Was this page helpful?

Provide product feedback  | Get help at Microsoft Q&A

# convert\_angle

Article • 03/26/2023

Convert an angle value from one unit to another.

## Syntax

```
convert_angle(value, from, to)
```

## Parameters

Name	Type	Required	Description
<code>value</code>	real	✓	The value to be converted.
<code>from</code>	string	✓	The unit to convert from. For possible values, see Conversion units.
<code>to</code>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- Arcminute
- Arcsecond
- Centiradian
- Deciradian
- Degree
- Gradian
- Microdegree
- Microradian
- Millidegree
- Milliradian
- Nanodegree
- Nanoradian
- NatoMil
- Radian
- Revolution
- Tilt

## Returns

Returns the input value converted from one angle unit to another. Invalid units return `null`.

## Example

[Run the query](#)

Kusto

```
print result = convert_angle(1.2, 'Degree', 'Arcminute')
```

## Output

result
72

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_energy

Article • 03/26/2023

Convert an energy value from one unit to another.

## Syntax

```
convert_energy(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- BritishThermalUnit
- Calorie
- DecathermEc
- DecathermImperial
- DecathermUs
- ElectronVolt
- Erg
- FootPound
- GigabritishThermalUnit
- GigaelectronVolt
- Gigajoule
- GigawattDay
- GigawattHour
- HorsepowerHour
- Joule
- KilobritishThermalUnit
- Kilocalorie
- KiloelectronVolt
- Kilojoule

- KilowattDay
- KilowattHour
- MegabritishThermalUnit
- Megacalorie
- MegaelectronVolt
- Megajoule
- MegawattDay
- MegawattHour
- Millijoule
- TeraelectronVolt
- TerawattDay
- TerawattHour
- ThermEc
- ThermImperial
- ThermUs
- WattDay
- WattHour

## Returns

Returns the input value converted from one energy unit to another. Invalid units return `null`.

## Example

Run the query

Kusto

```
print result = convert_energy(1.2, 'Joule', 'BritishThermalUnit')
```

## Output

**result**

```
0.00113738054437598
```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_force

Article • 03/26/2023

Convert a force value from one unit to another.

## Syntax

```
convert_force(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- Decanewton
- Dyn
- KilogramForce
- Kilonewton
- KiloPond
- KilopoundForce
- Meganewton
- Micronewton
- Millinewton
- Newton
- OunceForce
- Poundal
- PoundForce
- ShortTonForce
- TonneForce

## Returns

Returns the input value converted from one force unit to another. Invalid units return `null`.

## Example

[Run the query](#)

Kusto

```
print result = convert_force(1.2, 'Newton', 'Decanewton')
```

## Output

result
0.12

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_length

Article • 03/26/2023

Convert a length value from one unit to another.

## Syntax

```
convert_length(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- Angstrom
- AstronomicalUnit
- Centimeter
- Chain
- DataMile
- Decameter
- Decimeter
- DtpPica
- DtpPoint
- Fathom
- Foot
- Hand
- Hectometer
- Inch
- KilolightYear
- Kilometer
- Kiloparsec
- LightYear
- MegalightYear

- Megaparsec
- Meter
- Microinch
- Micrometer
- Mil
- Mile
- Millimeter
- Nanometer
- NauticalMile
- Parsec
- PrinterPica
- PrinterPoint
- Shackle
- SolarRadius
- Twip
- UsSurveyFoot
- Yard

## Returns

Returns the input value converted from one length unit to another. Invalid units return `null`.

## Example

[Run the query](#)

Kusto

```
print result = convert_length(1.2, 'Meter', 'Foot')
```

## Output

result
3.93700787401575

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_mass

Article • 03/26/2023

Convert a mass value from one unit to another.

## Syntax

```
convert_mass(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- Centigram
- Decagram
- Decigram
- EarthMass
- Grain
- Gram
- Hectogram
- Kilogram
- Kilopound
- Kilotonne
- LongHundredweight
- LongTon
- Megapound
- Megatonne
- Microgram
- Milligram
- Nanogram
- Ounce
- Pound

- ShortHundredweight
- ShortTon
- Slug
- SolarMass
- Stone
- Tonne

## Returns

Returns the input value converted from one mass unit to another. Invalid units return `null`.

## Example

[Run the query](#)

Kusto

```
print result = convert_mass(1.2, 'Kilogram', 'Pound')
```

## Output

**result**

```
2.64554714621853
```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_speed

Article • 03/26/2023

Convert a speed value from one unit to another.

## Syntax

```
convert_speed(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- CentimeterPerHour
- CentimeterPerMinute
- CentimeterPerSecond
- DecimeterPerMinute
- DecimeterPerSecond
- FootPerHour
- FootPerMinute
- FootPerSecond
- InchPerHour
- InchPerMinute
- InchPerSecond
- KilometerPerHour
- KilometerPerMinute
- KilometerPerSecond
- Knot
- MeterPerHour
- MeterPerMinute
- MeterPerSecond
- MicrometerPerMinute

- MicrometerPerSecond
- MilePerHour
- MillimeterPerHour
- MillimeterPerMinute
- MillimeterPerSecond
- NanometerPerMinute
- NanometerPerSecond
- UsSurveyFootPerHour
- UsSurveyFootPerMinute
- UsSurveyFootPerSecond
- YardPerHour
- YardPerMinute
- YardPerSecond

## Returns

Returns the input value converted from one speed unit to another. Invalid units return `null`.

## Example

Run the query

Kusto

```
print result = convert_speed(1.2, 'MeterPerSecond', 'CentimeterPerHour')
```

## Output

result
432000

## Feedback

Was this page helpful?

 Yes

 No

# convert\_temperature

Article • 03/26/2023

Convert a temperature value from one unit to another.

## Syntax

```
convert_temperature(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- DegreeCelsius
- DegreeDelisle
- DegreeFahrenheit
- DegreeNewton
- DegreeRankine
- DegreeReaumur
- DegreeRoemer
- Kelvin
- MillidegreeCelsius
- SolarTemperature

## Returns

Returns the input value converted from one temperature unit to another. Invalid units return `null`.

## Example

**Run the query**

Kusto

```
print result = convert_temperature(1.2, 'Kelvin', 'DegreeCelsius')
```

## Output

result
-271.95

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# convert\_volume

Article • 05/23/2023

Convert a volume value from one unit to another.

## Syntax

```
convert_volume(value, from, to)
```

## Parameters

Name	Type	Required	Description
<i>value</i>	real	✓	The value to be converted.
<i>from</i>	string	✓	The unit to convert from. For possible values, see Conversion units.
<i>to</i>	string	✓	The unit to convert to. For possible values, see Conversion units.

## Conversion units

- AcreFoot
- AuTablespoon
- BoardFoot
- Centiliter
- CubicCentimeter
- CubicDecimeter
- CubicFoot
- CubicHectometer
- CubicInch
- CubicKilometer
- CubicMeter
- CubicMicrometer
- CubicMile
- CubicMillimeter
- CubicYard
- Decaliter
- DecausGallon
- Deciliter
- DeciusGallon

- HectocubicFoot
- HectocubicMeter
- Hectoliter
- HectousGallon
- ImperialBeerBarrel
- ImperialGallon
- ImperialOunce
- ImperialPint
- KilocubicFoot
- KilocubicMeter
- KiloimperialGallon
- Kiloliter
- KilousGallon
- Liter
- MegacubicFoot
- MegaimperialGallon
- Megaliter
- MegausGallon
- MetricCup
- MetricTeaspoon
- Microliter
- Milliliter
- OilBarrel
- UkTablespoon
- UsBeerBarrel
- UsCustomaryCup
- UsGallon
- UsLegalCup
- UsOunce
- UsPint
- UsQuart
- UsTablespoon
- UsTeaspoon

## Returns

Returns the input value converted from one volume unit to another. Invalid units return `null`.

# Example

Run the query

Kusto

```
print result = convert_volume(1.2, 'CubicMeter', 'AcreFoot')
```

## Output

result
0.0009728568

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# cos()

Article • 03/06/2023

Returns the cosine function value of the specified angle. The angle is specified in radians.

## Syntax

```
cos(number)
```

## Parameters

Name	Type	Required	Description
<i>number</i>	real	✓	The value in radians for which to calculate the cosine.

## Returns

The cosine of *number* of radians.

## Example

Run the query

```
Kusto
```

```
print cos(1)
```

## Output

```
result
```

```
0.54030230586813977
```

---

## Feedback

Was this page helpful?

 Yes

 No

[Provide product feedback](#) | [Get help at Microsoft Q&A](#)

# cot()

Article • 12/28/2022

Calculates the trigonometric cotangent of the specified angle, in radians.

## Syntax

```
cot(number)
```

## Parameters

Name	Type	Required	Description
<i>number</i>	real	✓	The value for which to calculate the cotangent.

## Returns

The cotangent function value for *number*.

## Example

Run the query

```
Kusto
```

```
print cot(1)
```

## Output

```
result
```

```
0.64209261593433065
```

## Feedback

Was this page helpful?

 Yes

 No



# countof()

Article • 12/12/2022

Counts occurrences of a substring in a string. Plain string matches may overlap; regex matches don't.

## Syntax

```
countof(source, search [, kind])
```

## Parameters

Name	Type	Required	Description
<i>source</i>	string	✓	The value to search.
<i>search</i>	string	✓	The value or regular expression to match inside <i>source</i> .
<i>kind</i>	string		The value <code>normal</code> or <code>regex</code> . The default is <code>normal</code> .

## Returns

The number of times that the *search* value can be matched in the *source* string. Plain string matches may overlap; regex matches don't.

## Examples

Function call	Result
<code>countof("aaa", "a")</code>	3
<code>countof("aaaa", "aa")</code>	3 (not 2!)
<code>countof("ababa", "ab", "normal")</code>	2
<code>countof("ababa", "aba")</code>	2
<code>countof("ababa", "aba", "regex")</code>	1
<code>countof("abcabc", "a.c", "regex")</code>	2

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# current\_cluster\_endpoint()

Article • 11/23/2022

Returns the network endpoint (DNS name) of the current cluster being queried.

## Syntax

```
current_cluster_endpoint()
```

## Returns

The network endpoint (DNS name) of the current cluster being queried, as a value of type `string`.

## Example

Kusto

```
print strcat("This query executed on: ", current_cluster_endpoint())
```

## Feedback

Was this page helpful?



Yes



No

Provide product feedback | Get help at Microsoft Q&A

# current\_database()

Article • 11/23/2022

Returns the name of the database in scope (database that all query entities are resolved against if no other database is specified).

## Syntax

```
current_database()
```

## Returns

The name of the database in scope as a value of type `string`.

## Example

```
Kusto  
print strcat("Database in scope: ", current_database())
```

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# current\_principal()

Article • 04/18/2023

Returns the current principal name that runs the query.

## Syntax

```
current_principal()
```

## Returns

The current principal fully qualified name (FQN) as a `string`.

The string format is:

`PrincipalType=PrincipalId ; TenantId`

## Example

Run the query

Kusto

```
print fqdn=current_principal()
```

### Example output

`fqn`

```
aaduser=346e950e-4a62-42bf-96f5-4cf4eac3f11e;72f988bf-86f1-41af-91ab-2d7cd011db47
```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# current\_principal\_details()

Article • 04/18/2023

Returns details of the principal running the query.

## Syntax

```
current_principal_details()
```

## Returns

The details of the current principal as a `dynamic`.

## Example

Run the query

Kusto

```
print d=current_principal_details()
```

### Example output

```
d
{
  "UserPrincipalName": "user@fabrikam.com",
  "IdentityProvider": "https://sts.windows.net",
  "Authority": "72f988bf-86f1-41af-91ab-2d7cd011db47",
  "Mfa": "True",
  "Type": "AadUser",
  "DisplayName": "James Smith (upn: user@fabrikam.com)",
  "Objectid": "346e950e-4a62-42bf-96f5-4cf4eac3f11e",
  "FQN": null,
  "Notes": null
}
```

## Feedback

Was this page helpful?

Provide product feedback  | Get help at Microsoft Q&A

# current\_principal\_is\_member\_of()

Article • 04/21/2023

Checks group membership or principal identity of the current principal running the query.

## Syntax

```
current_principal_is_member_of(group)
```

## Parameters

Name	Type	Required	Description
<i>group</i>	dynamic	✓	An array of string literals in which each literal represents an Azure Active Directory (Azure AD) principal. See examples for Azure AD principals.

## Returns

The function returns `true` if the current principal running the query is successfully matched for at least one input argument. If not, the function returns `false`.

## Examples

Run the query

Kusto

```
print result=current_principal_is_member_of(
    'aaduser=user1@fabrikam.com',
    'aadgroup=group1@fabrikam.com',
    'aadapp=66ad1332-3a94-4a69-9fa2-17732f093664;72f988bf-86f1-41af-91ab-
2d7cd011db47'
)
```

## Output

**result**

**result**

false

Using dynamic array instead of multiple arguments:

**Run the query**

Kusto

```
print result=current_principal_is_member_of(
    dynamic([
        'aaduser=user1@fabrikam.com',
        'aadgroup=group1@fabrikam.com',
        'aadapp=66ad1332-3a94-4a69-9fa2-17732f093664;72f988bf-86f1-41af-91ab-
2d7cd011db47'
    ]))
```

**Output**

**result**

false

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# cursor\_after()

Article • 12/12/2022

A predicate run over the records of a table to compare their ingestion time against a database cursor.

## ⓘ Note

This function can only be invoked on records of a table that has the **IngestionTime policy** enabled.

## Syntax

```
cursor_after(RHS)
```

## Parameters

Name	Type	Required	Description
<i>RHS</i>	string	✓	Either an empty string literal or a valid database cursor value.

## Returns

A scalar value of type `bool` that indicates whether the record was ingested after the database cursor *RHS* (`true`) or not (`false`).

## See also

See database cursors for additional details on database cursors.

## Feedback

Was this page helpful?

 Yes

 No

# cursor\_before\_or\_at()

Article • 12/12/2022

A predicate function run over the records of a table to compare their ingestion time against the database cursor time.

## ⓘ Note

This function can only be invoked on records of a table that has the **IngestionTime policy** enabled.

## Syntax

```
cursor_before_or_at(RHS)
```

## Parameters

Name	Type	Required	Description
<i>RHS</i>	string	✓	Either an empty string literal or a valid database cursor value.

## Returns

A scalar value of type `bool` that indicates whether the record was ingested before or at the database cursor *RHS* (`true`) or not (`false`).

## See also

See database cursors for additional details on database cursors.

## Feedback

Was this page helpful?

👍 Yes

👎 No

# cursor\_current()

Article • 11/08/2022

Retrieves the current value of the cursor of the database in scope.

Deprecated aliases: current\_cursor()

## Syntax

```
cursor_current()
```

## Returns

Returns a single value of type `string` that encodes the current value of the cursor of the database in scope.

### Notes

See database cursors for more details on database cursors.

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

## **datetime\_add()**

Article • 03/22/2023

Calculates a new datetime from a specified period multiplied by a specified amount, added to, or subtracted from a specified datetime.

## Syntax

```
datetime add(period, amount, datetime)
```

## Parameters

Name	Type	Required	Description
<i>period</i>	string	✓	The length of time by which to increment.
<i>amount</i>	int	✓	The number of <i>periods</i> to add to or subtract from <i>datetime</i> .
<i>datetime</i>	datetime	✓	The date to increment by the result of the <i>period</i> x <i>amount</i> calculation.

Possible values of *period*:

- Year
  - Quarter
  - Month
  - Week
  - Day
  - Hour
  - Minute
  - Second
  - Millisecond
  - Microsecond
  - Nanosecond

## Returns

A datetime after a certain time/date interval has been added.

## Examples

## Period

## Run the query

Kusto

```
print year = datetime_add('year',1,make_datetime(2017,1,1)),
quarter = datetime_add('quarter',1,make_datetime(2017,1,1)),
month = datetime_add('month',1,make_datetime(2017,1,1)),
week = datetime_add('week',1,make_datetime(2017,1,1)),
day = datetime_add('day',1,make_datetime(2017,1,1)),
hour = datetime_add('hour',1,make_datetime(2017,1,1)),
minute = datetime_add('minute',1,make_datetime(2017,1,1)),
second = datetime_add('second',1,make_datetime(2017,1,1))
```

## Output

**year**      **quarter**      **month**      **week**      **day**      **hour**      **minute**      **second**

year	quarter	month	week	day	hour	minute	second
2018-01-01 00:00:00.0000000	2017-04-01 00:00:00.0000000	2017-02-01 00:00:00.0000000	2017-01-08 00:00:00.0000000	2017-01-02 00:00:00.0000000	2017-01-01 01:00:00.0000000	2017-01-01 00:01:00.0000000	2017-01-01 00:00:01.0000000

## Amount

[Run the query](#)

Kusto

```
print year = datetime_add('year', -5, make_datetime(2017,1,1)),
       quarter = datetime_add('quarter', 12, make_datetime(2017,1,1)),
       month = datetime_add('month', -15, make_datetime(2017,1,1)),
       week = datetime_add('week', 100, make_datetime(2017,1,1))
```

## Output

year	quarter	month	week
2012-01-01T00:00:00Z	2020-01-01T00:00:00Z	2015-10-01T00:00:00Z	2018-12-02T00:00:00Z

---

## Feedback

Was this page helpful?

Provide product feedback [↗](#) | Get help at Microsoft Q&A

# datetime\_diff()

Article • 12/28/2022

Calculates the number of the specified periods between two datetime values.

## Syntax

```
datetime_diff(period, datetime1, datetime2)
```

## Parameters

Name	Type	Required	Description
<i>period</i>	string	✓	The measurement of time used to calculate the return value. See possible values.
<i>datetime1</i>	datetime	✓	The left-hand side of the subtraction equation.
<i>datetime2</i>	datetime	✓	The right-hand side of the subtraction equation.

## Possible values of *period*

- Year
- Quarter
- Month
- Week
- Day
- Hour
- Minute
- Second
- Millisecond
- Microsecond
- Nanosecond

## Returns

An integer that represents the amount of *periods* in the result of subtraction (*datetime1* - *datetime2*).

## Example

Run the query

```
print
year = datetime_diff('year',datetime(2017-01-01),datetime(2000-12-31)),
quarter = datetime_diff('quarter',datetime(2017-07-01),datetime(2017-03-30)),
month = datetime_diff('month',datetime(2017-01-01),datetime(2015-12-30)),
week = datetime_diff('week',datetime(2017-10-29 00:00),datetime(2017-09-30 23:59)),
day = datetime_diff('day',datetime(2017-10-29 00:00),datetime(2017-09-30 23:59)),
hour = datetime_diff('hour',datetime(2017-10-31 01:00),datetime(2017-10-30 23:59)),
minute = datetime_diff('minute',datetime(2017-10-30 23:05:01),datetime(2017-10-30 23:00:59)),
second = datetime_diff('second',datetime(2017-10-30 23:00:10.100),datetime(2017-10-30 23:00:00.900)),
millisecond = datetime_diff('millisecond',datetime(2017-10-30 23:00:00.200100),datetime(2017-10-30 23:00:00.100900)),
microsecond = datetime_diff('microsecond',datetime(2017-10-30 23:00:00.1009001),datetime(2017-10-30 23:00:00.1008009)),
nanosecond = datetime_diff('nanosecond',datetime(2017-10-30 23:00:00.0000000),datetime(2017-10-30 23:00:00.000007))
```

## Output

year	quarter	month	week	day	hour	minute	second	millisecond	microsecond	nanosecond
17	2	13	5	29	2	5	10	100	100	-700

## Feedback

Was this page helpful?



Provide product feedback [↗](#) | Get help at Microsoft Q&A

# datetime\_local\_to\_utc()

Article • 12/28/2022

Converts local datetime to UTC datetime using a time-zone specification.

## Syntax

```
datetime_local_to_utc(from, timezone)
```

## Parameters

Name	Type	Required	Description
<i>from</i>	datetime	✓	The local datetime to convert.
<i>timezone</i>	string	✓	The timezone of the desired datetime. The value must be one of the supported timezones.

## Returns

A UTC datetime that corresponds the local datetime in the specified `timezone`.

## Example

Run the query

```
Kusto

datatable(local_dt: datetime, tz: string)
[ datetime(2020-02-02 20:02:20), 'US/Pacific',
  datetime(2020-02-02 20:02:20), 'America/Chicago',
  datetime(2020-02-02 20:02:20), 'Europe/Paris']
| extend utc_dt = datetime_local_to_utc(local_dt, tz)
```

## Output

local_dt	tz	utc_dt
2020-02-02 20:02:20.0000000	Europe/Paris	2020-02-02 19:02:20.0000000
2020-02-02 20:02:20.0000000	America/Chicago	2020-02-03 02:02:20.0000000

<b>local_dt</b>	<b>tz</b>	<b>utc_dt</b>
2020-02-02 20:02:20.0000000	US/Pacific	2020-02-03 04:02:20.0000000

### ⓘ Note

Normally there is a 1:1 mapping between UTC and local time, however there is a time ambiguity near the DST transition. Translating from local to UTC and then back to local may produce an hour offset between two local datetime values if the clocks were advanced due to DST.

Kusto

```
range Local from datetime(2022-03-27 01:00:00.000000) to datetime(2022-03-27 04:00:00.000000) step 1h
| extend UTC=datetime_local_to_utc(Local, 'Europe/Brussels')
| extend BackToLocal=datetime_utc_to_local(UTC, 'Europe/Brussels')
| extend diff=Local-BackToLocal
```

<b>Local</b>	<b>UTC</b>	<b>BackToLocal</b>	<b>diff</b>
2022-03-27 02:00:00.0000000	2022-03-27 00:00:00.0000000	2022-03-27 01:00:00.0000000	01:00:00
2022-03-27 01:00:00.0000000	2022-03-27 00:00:00.0000000	2022-03-27 01:00:00.0000000	00:00:00
2022-03-27 03:00:00.0000000	2022-03-27 01:00:00.0000000	2022-03-27 03:00:00.0000000	00:00:00
2022-03-27 04:00:00.0000000	2022-03-27 02:00:00.0000000	2022-03-27 04:00:00.0000000	00:00:00

## See also

- To convert from UTC to local, see `datetime_utc_to_local()`
- List of supported timezones

---

## Feedback

Was this page helpful?

Yes

No



# datetime\_part()

Article • 12/28/2022

Extracts the requested date part as an integer value.

Deprecated aliases: datepart()

## Syntax

```
datetime_part(part, datetime)
```

## Parameters

Name	Type	Required	Description
part	string	✓	Measurement of time to extract from <i>date</i> . See possible values.
date	datetime	✓	The full date from which to extract <i>part</i> .

## Possible values of part

- Year
- Quarter
- Month
- week\_of\_year
- Day
- DayOfYear
- Hour
- Minute
- Second
- Millisecond
- Microsecond
- Nanosecond

## Returns

An integer representing the extracted part.

ⓘ Note

`week_of_year` returns an integer which represents the week number. The week number is calculated from the first week of a year, which is the one that includes the first Thursday.

## Example

Run the query

Kusto

```
let dt = datetime(2017-10-30 01:02:03.7654321);
print
year = datetime_part("year", dt),
quarter = datetime_part("quarter", dt),
month = datetime_part("month", dt),
weekOfYear = datetime_part("week_of_year", dt),
day = datetime_part("day", dt),
dayOfYear = datetime_part("dayOfYear", dt),
hour = datetime_part("hour", dt),
minute = datetime_part("minute", dt),
second = datetime_part("second", dt),
millisecond = datetime_part("millisecond", dt),
microsecond = datetime_part("microsecond", dt),
nanosecond = datetime_part("nanosecond", dt)
```

## Output

year	quarter	month	weekOfYear	day	dayOfYear	hour	minute	second	millisecond	microsecond	nanosecond
2017	4	10	44		30	303	1	2	3	765	765432

### ⓘ Note

`weekofyear` is an obsolete variant of `week_of_year` part. `weekofyear` was not ISO 8601 compliant; the first week of a year was defined as the week with the year's first Wednesday in it. `week_of_year` is ISO 8601 compliant; the first week of a year is defined as the week with the year's first Thursday in it. [For more information], see [ISO 8601 week dates](#).

## Feedback

Was this page helpful?

Provide product feedback | Get help at Microsoft Q&A

# datetime\_utc\_to\_local()

Article • 12/28/2022

Converts UTC datetime to local datetime using a time-zone specification.

## Syntax

```
datetime_utc_to_local(from, timezone)
```

## Parameters

Name	Type	Required	Description
<i>from</i>	datetime	✓	The UTC datetime to convert.
<i>timezone</i>	string	✓	The timezone to convert to. This value must be one of the supported timezones.

## Returns

A local datetime in the *timezone* that corresponds the UTC datetime.

## Example

Run the query

Kusto

```
print dt=now()
| extend pacific_dt = datetime_utc_to_local(dt, 'US/Pacific'), canberra_dt =
datetime_utc_to_local(dt, 'Australia/Canberra')
| extend diff = pacific_dt - canberra_dt
```

## Output

dt	pacific_dt	canberra_dt	diff
2022-07-11 22:18:48.4678620	2022-07-11 15:18:48.4678620	2022-07-12 08:18:48.4678620	-17:00:00

## See also

- To convert a datetime from local to UTC, see `datetime_local_to_utc()`.
  - Timezones
- 

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# dayofmonth()

Article • 12/28/2022

Returns an integer representing the day number of the given datetime.

## Syntax

```
dayofmonth( date )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The datetime used to extract the day number.

## Returns

An integer representing the day number of the given datetime.

## Example

Run the query

```
Kusto  
dayofmonth(datetime('2015-12-14'))
```

## Output

result
14

## Feedback

Was this page helpful?

 Yes

 No



# dayofweek()

Article • 12/12/2022

Returns the integer number of days since the preceding Sunday, as a `timespan`.

## Syntax

```
dayofweek( date )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The datetime for which to determine the day of week.

## Returns

The `timespan` since midnight at the beginning of the preceding Sunday, rounded down to an integer number of days.

## Examples

Kusto

```
dayofweek(datetime(1947-11-30 10:00:05)) // time(0.00:00:00), indicating  
Sunday  
dayofweek(datetime(1970-05-11)) // time(1.00:00:00), indicating  
Monday
```

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# dayofyear()

Article • 12/28/2022

Returns the integer number represents the day number of the given year.

## Syntax

```
dayofyear( date )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The datetime for which to determine the day number.

## Returns

The day number of the given year.

## Example

Run the query

Kusto

```
dayofyear(datetime('2015-12-14'))
```

## Output

**result**

348

## Feedback

Was this page helpful?

 Yes

 No



# dcount\_hll()

Article • 12/28/2022

Calculates the distinct count from results generated by hll or hll\_merge.

Read about the underlying algorithm (*HyperLogLog*) and estimation accuracy.

## Syntax

```
dcount_hll(hll)
```

## Parameters

Name	Type	Required	Description
hll	string	✓	An expression generated by hll or hll-merge to be used to find the distinct count.

## Returns

Returns the distinct count of each value in *hll*.

## Example

The following example shows the distinct count hll merged results.

Run the query

Kusto

```
StormEvents
| summarize hllRes = hll(DamageProperty) by bin(StartTime,10m)
| summarize hllMerged = hll_merge(hllRes)
| project dcount_hll(hllMerged)
```

## Output

**dcount\_hll\_hllMerged**

315

# Estimation accuracy

This function uses a variant of the HyperLogLog (HLL) algorithm<sup>↗</sup>, which does a stochastic estimation of set cardinality. The algorithm provides a "knob" that can be used to balance accuracy and execution time per memory size:

Accuracy	Error (%)	Entry count
0	1.6	$2^{12}$
1	0.8	$2^{14}$
2	0.4	$2^{16}$
3	0.28	$2^{17}$
4	0.2	$2^{18}$

## ⓘ Note

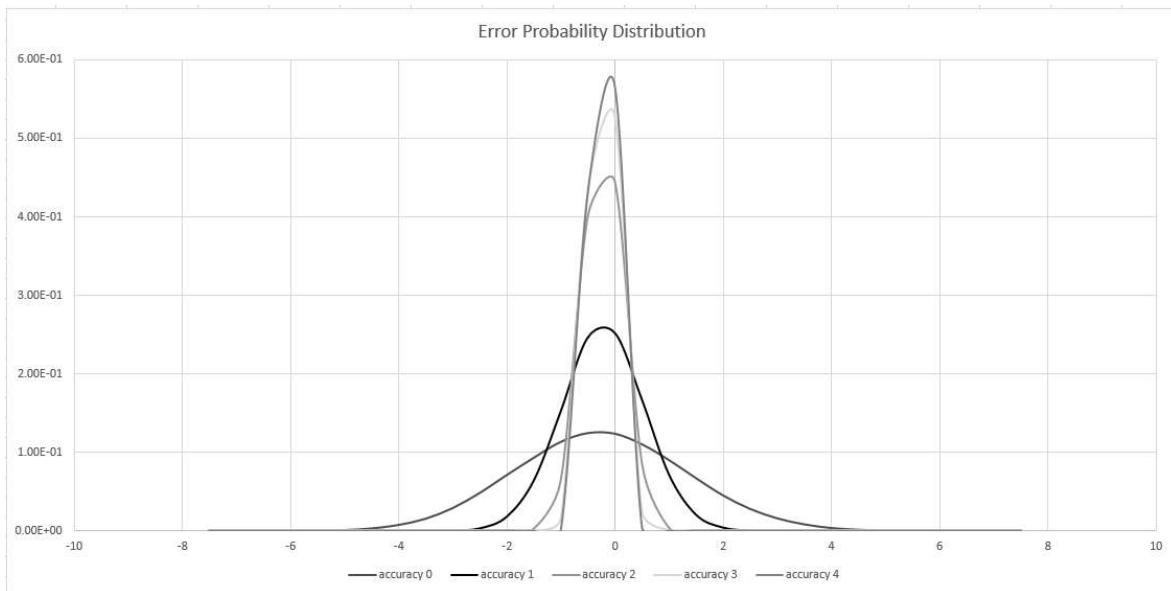
The "entry count" column is the number of 1-byte counters in the HLL implementation.

The algorithm includes some provisions for doing a perfect count (zero error), if the set cardinality is small enough:

- When the accuracy level is 1, 1000 values are returned
- When the accuracy level is 2, 8000 values are returned

The error bound is probabilistic, not a theoretical bound. The value is the standard deviation of error distribution (the sigma), and 99.7% of the estimations will have a relative error of under  $3 \times \sigma$ .

The following image shows the probability distribution function of the relative estimation error, in percentages, for all supported accuracy settings:



---

## Feedback

Was this page helpful?

Provide product feedback [↗](#) | Get help at Microsoft Q&A

# degrees()

Article • 12/28/2022

Converts angle value in radians into value in degrees, using the formula `degrees = (180 / PI) * angle_in_radians.`

## Syntax

```
degrees(radians)
```

## Parameters

Name	Type	Required	Description
<i>radians</i>	real	✓	The angle in radians to convert to degrees.

## Returns

The corresponding angle in degrees for an angle specified in radians.

## Examples

Run the query

Kusto

```
print degrees0 = degrees(pi()/4), degrees1 = degrees(pi()*1.5), degrees2 = degrees(0)
```

## Output

degrees0	degrees1	degrees2
45	270	0

## Feedback



Was this page helpful?

Provide product feedback  | Get help at Microsoft Q&A

# dynamic\_to\_json()

Article • 03/09/2023

Converts a scalar value of type `dynamic` to a canonical `string` representation.

## Syntax

```
dynamic_to_json(expr)
```

## Parameters

Name	Type	Required	Description
<i>expr</i>	<code>dynamic</code>	✓	The expression to convert to string representation.

## Returns

Returns a canonical representation of the input as a value of type `string`, according to the following rules:

- If the input is a scalar value of type other than `dynamic`, the output is the application of `tostring()` to that value.
- If the input is an array of values, the output is composed of the characters `[`, `,`, and `]` interspersed with the canonical representation described here of each array element.
- If the input is a property bag, the output is composed of the characters `{`, `,`, and `}` interspersed with the colon (`:`)-delimited name/value pairs of the properties. The pairs are sorted by the names, and the values are in the canonical representation described here of each array element.

## Example

Run the query

Kusto

```

let bag1 = dynamic_to_json(
    dynamic({
        'Y10':dynamic({}),
        'X8': dynamic({
            'c3':1,
            'd8':5,
            'a4':6
        }),
        'D1':114,
        'A1':12,
        'B1':2,
        'C1':3,
        'A14':[15, 13, 18]
    }));
let bag2 = dynamic_to_json(
    dynamic({
        'X8': dynamic({
            'a4':6,
            'c3':1,
            'd8':5
        }),
        'A14':[15, 13, 18],
        'C1':3,
        'B1':2,
        'Y10': dynamic({}),
        'A1':12, 'D1':114
    }));
print AreEqual=bag1 == bag2, Result=bag1

```

## Output

AreEqual	Result
true	{"A1":12,"A14":[15,13,18],"B1":2,"C1":3,"D1":114,"X8":{"a4":6,"c3":1,"d8":5},"Y10":{}}

## Feedback

Was this page helpful?

Provide product feedback  | Get help at Microsoft Q&A

# endofday()

Article • 12/28/2022

Returns the end of the day containing the date, shifted by an offset, if provided.

## Syntax

```
endofday( date [, offset] )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The date to find the end of.
<i>offset</i>	int		The number of offset days from <i>date</i> . Default is 0.

## Returns

A datetime representing the end of the day for the given *date* value, with the *offset*, if specified.

## Example

[Run the query](#)

Kusto

```
range offset from -1 to 1 step 1
| project dayEnd = endofday(datetime(2017-01-01 10:10:17), offset)
```

## Output

dayEnd
2016-12-31 23:59:59.9999999
2017-01-01 23:59:59.9999999
2017-01-02 23:59:59.9999999

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# endofmonth()

Article • 12/28/2022

Returns the end of the month containing the date, shifted by an offset, if provided.

## Syntax

```
endofmonth( date [, offset] )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The date used to find the end of the month.
<i>offset</i>	int		The number of offset months from <i>date</i> . Default is 0.

## Returns

A datetime representing the end of the month for the given *date* value, with the *offset*, if specified.

## Example

Run the query

Kusto

```
range offset from -1 to 1 step 1
| project monthEnd = endofmonth(datetime(2017-01-01 10:10:17), offset)
```

## Output

monthEnd
2016-12-31 23:59:59.9999999
2017-01-31 23:59:59.9999999
2017-02-28 23:59:59.9999999

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# endofweek()

Article • 12/28/2022

Returns the end of the week containing the date, shifted by an offset, if provided.

Last day of the week is considered to be a Saturday.

## Syntax

```
endofweek( date [, offset] )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The date used to find the end of the week.
<i>offset</i>	int		The number of offset weeks from <i>date</i> . Default is 0.

## Returns

A datetime representing the end of the week for the given *date* value, with the *offset*, if specified.

## Example

Run the query

```
Kusto
```

```
range offset from -1 to 1 step 1
| project weekEnd = endofweek(datetime(2017-01-01 10:10:17), offset)
```

## Output

weekEnd
2016-12-31 23:59:59.9999999
2017-01-07 23:59:59.9999999

**weekEnd**

2017-01-14 23:59:59.9999999

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# endofyear()

Article • 12/28/2022

Returns the end of the year containing the date, shifted by an offset, if provided.

## Syntax

```
endofyear( date [, offset] )
```

## Parameters

Name	Type	Required	Description
<i>date</i>	datetime	✓	The date used to find the end of the year.
<i>offset</i>	int		The number of offset years from <i>date</i> . Default is 0.

## Returns

A datetime representing the end of the year for the given *date* value, with the &*offset*, if specified.

## Example

Run the query

Kusto

```
range offset from -1 to 1 step 1
| project yearEnd = endofyear(datetime(2017-01-01 10:10:17), offset)
```

## Output

yearEnd
2016-12-31 23:59:59.9999999
2017-12-31 23:59:59.9999999
2018-12-31 23:59:59.9999999

---

# Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# estimate\_data\_size()

Article • 05/30/2023

Returns an estimated data size in bytes of the selected columns of the tabular expression.

## Syntax

```
estimate_data_size(columns)
```

## Parameters

Name	Type	Required	Description
<i>columns</i>	string	✓	One or more comma-separated column references in the source tabular expression to use for data size estimation. To include all columns, use the wildcard (*) character.

## Returns

The estimated data size in bytes of the record size. Estimation is based on data types and values lengths.

## Example

The following example calculates the total data size using `estimate_data_size()`.

Run the query

Kusto

```
range x from 1 to 10 step 1          // x (long) is 8
| extend Text = '1234567890'        // Text length is 10
| summarize Total=sum(estimate_data_size(*)) // (8+10)x10 = 180
```

## Output

Total
180

## See also

- Estimate table size
- 

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# exp()

Article • 12/28/2022

The base-e exponential function of x, which is e raised to the power x:  $e^x$ .

## Syntax

```
exp(x)
```

## Parameters

Name	Type	Required	Description
x	real	✓	The value of the exponent.

## Returns

The exponential value of x.

## See also

- For natural (base-e) logarithms, see [log\(\)](#).
- For exponential functions of base-2 and base-10 logarithms, see [exp2\(\)](#), [exp10\(\)](#).

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# exp10()

Article • 12/28/2022

The base-10 exponential function of x, which is 10 raised to the power x:  $10^x$ .

## Syntax

```
exp10(x)
```

## Parameters

Name	Type	Required	Description
x	real	✓	The value of the exponent.

## Returns

The exponential value of x.

## See also

- For natural (base-10) logarithms, see [log10\(\)](#).
- For exponential functions of base-e and base-2 logarithms, see [exp\(\)](#), [exp2\(\)](#).

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# exp2()

Article • 12/28/2022

The base-2 exponential function of x, which is 2 raised to the power x:  $2^x$ .

## Syntax

```
exp2(x)
```

## Parameters

Name	Type	Required	Description
x	real	✓	The value of the exponent.

## Returns

The exponential value of x.

## See also

- For natural (base-2) logarithms, see [log2\(\)](#).
- For exponential functions of base-e and base-10 logarithms, see [exp\(\)](#), [exp10\(\)](#).

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# extent\_id()

Article • 12/13/2022

Returns a unique identifier that identifies the data shard ("extent") that the current record resides in.

Applying this function to calculated data that isn't attached to a data shard returns an empty guid (all zeros).

**Deprecated aliases:** extentid()

## Syntax

```
extent_id()
```

## Returns

A value of type `guid` that identifies the current record's data shard, or an empty guid (all zeros).

## Example

The following example shows how to get a list of all the data shards that have records from an hour ago with a specific value for the column `ActivityId`. It demonstrates that some query operators (here, the `where` operator, and also `extend` and `project`) preserve the information about the data shard hosting the record.

Kusto

```
T
| where Timestamp > ago(1h)
| where ActivityId == 'dd0595d4-183e-494e-b88e-54c52fe90e5a'
| extend eid=extent_id()
| summarize by eid
```

## Feedback

Was this page helpful?

 Yes

 No



# extent\_tags()

Article • 12/13/2022

Returns a dynamic array with the tags of the data shard ("extent") that the current record is in.

If you apply this function to calculated data, which isn't attached to a data shard, returns an empty value.

## Syntax

```
extent_tags()
```

## Returns

A value of type `dynamic` that is an array holding the current record's extent tags, or an empty value.

## Examples

Some query operators preserve the information about the data shard hosting the record. These operators include `where`, `extend`, and `project`. The following example shows how to get a list the tags of all the data shards that have records from an hour ago, with a specific value for the column `ActivityId`.

```
Kusto  
  
T  
| where Timestamp > ago(1h)  
| where ActivityId == 'dd0595d4-183e-494e-b88e-54c52fe90e5a'  
| extend tags = extent_tags()  
| summarize by tostring(tags)
```

The following example shows how to obtain a count of all records from the last hour, which are stored in extents tagged with the tag `MyTag` (and potentially other tags), but not tagged with the tag `drop-by:MyOtherTag`.

```
Kusto  
  
T  
| where Timestamp > ago(1h)
```

```
| extend Tags = extent_tags()  
| where Tags has_cs 'MyTag' and Tags !has_cs 'drop-by:MyOtherTag'  
| count
```

ⓘ Note

Filtering on the value of `extent_tags()` performs best when one of the following string operators is used: `has`, `has_cs`, `!has`, `!has_cs`.

## Feedback

Was this page helpful?



Yes



No

Provide product feedback ↗ | Get help at Microsoft Q&A

# extract()

Article • 12/28/2022

Get a match for a regular expression from a source string.

Optionally, convert the extracted substring to the indicated type.

## Syntax

```
extract(regex, captureGroup, source [, typeLiteral])
```

## Parameters

Name	Type	Required	Description
<i>regex</i>	string	✓	A regular expression.
<i>captureGroup</i>	int	✓	The capture group to extract. 0 stands for the entire match, 1 for the value matched by the first '(' parenthesis ')' in the regular expression, and 2 or more for subsequent parentheses.
<i>source</i>	string	✓	The string to search.
<i>typeLiteral</i>	string		If provided, the extracted substring is converted to this type. For example, <code>typeof(long)</code> .

## Returns

If *regex* finds a match in *source*: the substring matched against the indicated capture group *captureGroup*, optionally converted to *typeLiteral*.

If there's no match, or the type conversion fails: `null`.

## Examples

The example string `Trace` is searched for a definition for `Duration`. The match is converted to `real`, then multiplied it by a time constant (`1s`) so that `Duration` is of type `timespan`. In this example, it's equal to 123.45 seconds:

Kusto

```
...
| extend Trace="A=1, B=2, Duration=123.45, ..."
| extend Duration = extract("Duration=([0-9.]+)", 1, Trace, typeof(real)) *
time(1s)
```

This example is equivalent to `substring(Text, 2, 4)`:

Kusto

```
extract("^.{2,2}(.{4,4})", 1, Text)
```

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# extract\_all()

Article • 12/28/2022

Get all matches for a regular expression from a source string. Optionally, retrieve a subset of matching groups.

Kusto

```
print extract_all(@"(\d+)", "a set of numbers: 123, 567 and 789") // results with the dynamic array ["123", "567", "789"]
```

**Deprecated aliases:** extractall()

## Syntax

```
extract_all(regex, [captureGroups,] source)
```

## Parameters

Name	Type	Required	Description
<i>regex</i>	string	✓	A regular expression containing between one and 16 capture groups.
<i>captureGroups</i>	dynamic		An array that indicates the capture groups to extract. Valid values are from 1 to the number of capturing groups in the regular expression. Named capture groups are allowed as well. See examples.
<i>source</i>	string	✓	The string to search.

## Returns

- If *regex* finds a match in *source*: Returns dynamic array including all matches against the indicated capture groups *captureGroups*, or all of capturing groups in the *regex*.
- If number of *captureGroups* is 1: The returned array has a single dimension of matched values.
- If number of *captureGroups* is more than 1: The returned array is a two-dimensional collection of multi-value matches per *captureGroups* selection, or all capture groups present in the *regex* if *captureGroups* is omitted.

- If there's no match: `null`.

## Examples

### Extract a single capture group

The following query returns hex-byte representation (two hex-digits) of the GUID.

[Run the query](#)

Kusto

```
print Id="82b8be2d-dfa7-4bd1-8f63-24ad26d31449"
| extend guid_bytes = extract_all(@"([\da-f]{2})", Id)
```

#### Output

ID	guid_bytes
82b8be2d-dfa7-4bd1-8f63-24ad26d31449	[{"82","b8","be","2d","df","a7","4b","d1","8f","63","24","ad","26","d3","14","49"]

### Extract several capture groups

The following query uses a regular expression with three capturing groups to split each GUID part into first letter, last letter, and whatever is in the middle.

[Run the query](#)

Kusto

```
print Id="82b8be2d-dfa7-4bd1-8f63-24ad26d31449"
| extend guid_bytes = extract_all(@"(\w)(\w+)(\w)", Id)
```

#### Output

ID	guid_bytes
82b8be2d-dfa7-4bd1-8f63-24ad26d31449	[["8","2b8be2","d"], ["d","fa","7"], ["4","bd","1"], ["8","f6","3"], ["2","4ad26d3144","9"]]

## Extract a subset of capture groups

The following query selects a subset of capturing groups.

The regular expression matches the first letter, last letter, and all the rest.

The *captureGroups* parameter is used to select only the first and the last parts.

**Run the query**

Kusto

```
print Id="82b8be2d-dfa7-4bd1-8f63-24ad26d31449"
| extend guid_bytes = extract_all(@"(\w)(\w+)(\w)", dynamic([1,3]), Id)
```

**Output**

**ID**

**guid\_bytes**

82b8be2d-dfa7-4bd1-8f63-24ad26d31449

[["8","d"],["d","7"],["4","1"],["8","3"],["2","9"]]

## Using named capture groups

The *captureGroups* in the following query uses both capture group indexes and named capture group references to fetch matching values.

**Run the query**

Kusto

```
print Id="82b8be2d-dfa7-4bd1-8f63-24ad26d31449"
| extend guid_bytes = extract_all(@"(?P<first>\w)(?P<middle>\w+)(?P<last>\w)", dynamic(['first',2,'last']), Id)
```

**Output**

**ID**

**guid\_bytes**

82b8be2d-dfa7-4bd1-8f63-24ad26d31449

[["8","2b8be2","d"],["d","fa","7"],["4","bd","1"],["8","f6","3"],["2","4ad26d3144","9"]]

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# extract\_json()

Article • 03/12/2023

Get a specified element out of a JSON text using a path expression.

Optionally convert the extracted string to a specific type.

The `extract_json()` and `extractjson()` functions are equivalent

Kusto

```
extract_json("$.hosts[1].AvailableMB", EventText, typeof(int))
```

## Syntax

```
extract_json(jsonPath, dataSource, type)
```

## Parameters

Name	Type	Required	Description
<i>jsonPath</i>	string	✓	A JSONPath that defines an accessor into the JSON document.
<i>dataSource</i>	string	✓	A JSON document.
<i>type</i>	string		An optional type literal. If provided, the extracted value is converted to this type. For example, <code>typeof(long)</code> will convert the extracted value to a <code>long</code> .

## Performance tips

- Apply where-clauses before using `extract_json()`.
- Consider using a regular expression match with `extract` instead. This can run very much faster, and is effective if the JSON is produced from a template.
- Use `parse_json()` if you need to extract more than one value from the JSON.
- Consider having the JSON parsed at ingestion by declaring the type of the column to be dynamic.

## Returns

This function performs a JSONPath query into dataSource, which contains a valid JSON string, optionally converting that value to another type depending on the third argument.

## Example

The [ bracket ] notation and dot ( . ) notation are equivalent:

```
Kusto
```

```
T  
| extend AvailableMB = extract_json("$.hosts[1].AvailableMB", EventText,  
typeof(int))  
  
T  
| extend AvailableMB = extract_json("${['hosts'][1]['AvailableMB']}",  
EventText, typeof(int))
```

---

## Feedback

Was this page helpful?

 Yes

 No

Provide product feedback  | Get help at Microsoft Q&A

# format\_bytes()

Article • 05/30/2023

Formats a number as a string representing data size in bytes.

## Syntax

```
format_bytes(size [, precision [, units]])
```

## Parameters

Name	Type	Required	Description
<i>size</i>	real	✓	The value to be formatted as data size in bytes.
<i>precision</i>	int		The number of digits the value will be rounded to after the decimal point. The default is 0.
<i>units</i>	string		The units of the target data size: Bytes, KB, MB, GB, TB, or PB. If this parameter is empty, the units will be auto-selected based on input value.

## Returns

A string of *size* formatted as data size in bytes.

## Examples

Run the query

```
Kusto

print
v1 = format_bytes(564),
v2 = format_bytes(10332, 1),
v3 = format_bytes(20010332),
v4 = format_bytes(20010332, 2),
v5 = format_bytes(20010332, 0, "KB")
```

## Output