Date.Day

Returns the day for a DateTime value.
Syntax: Date.Day(dateTime as any) as nullable number
Example: Date.Day(#datetime(2011, 5, 14, 17, 0, 0))
Answer: 14

Date.DayOfWeek

Returns a number (from 0 to 6) indicating the day of the week of the provided value.
Syntax: Date.DayOfWeek(dateTime as any, optional firstDayOfWeek as nullable number [varies according to culture]) as nullable number
Example: Date.DayOfWeek(#date(2011, 02, 21), Day.Monday)
Answer: 0 Monday

Date.DayOfWeekName

Returns the day of the week name.
Syntax: Date.DayOfWeekName(date as any, optional culture as nullable text)
Example: Date.DayOfWeekName(#date(2011, 12, 31), "en-US")

Date.DayOfYear

Returns a number that represents the day of the year from a DateTime value.
Syntax: Date.DayOfYear(dateTime as any) as nullable number
Example: Date.DayOfYear(#date(2011, 03, 01))
Answer: 60

Date.DaysInMonth

Returns the number of days in the month from a DateTime value.
Syntax: Date.DaysInMonth(dateTime as any) as nullable number
Example: Date.DaysInMonth(#date(2011, 12, 01))
Answer: 31

Date.EndOfDay

Returns a DateTime value for the end of the day.
Example: Date.EndOfDay(#datetime(2011, 5, 14, 17, 0, 0))
Answer: #datetime(2011, 5, 14, 23, 59, 59.9999999)
Example: Date.EndOfDay(#datetimezone(2011, 5, 17, 5, 0, 0, -7, 0))
Answer: #datetimezone(2011, 5, 17, 23, 59, 59.9999999, -7, 0)

Date.EndOfMonth

Returns a DateTime value for the end of the month.
Syntax: Date.EndOfMonth(dateTime as any) as any
Example: Date.EndOfMonth(#date(2011, 5, 14))

Date.EndOfQuarter

Returns a Date/DateTime/DateTimeZone value representing the end of the quarter. The date and time portions are reset to their terminating values for the quarter. The timezone information is persisted.
Syntax: Date.EndOfQuarter(dateTime as any) as any
Example: Date.EndOfQuarter(#datetime(2011, 10, 10, 8, 0, 0))

Date.EndOfWeek

Returns a DateTime value for the end of the week.
Syntax: Date.EndOfWeek(dateTime as any, optional firstDayOfWeek as nullable number [Day.Sunday]) as any
Example: Date.EndOfWeek(#date(2011, 5, 14))
Answer: #date(2011, 5, 14)

Example: Date.EndOfWeek(#datetimezone(2011, 5, 17, 5, 0, 0, -7, 0), Day.Sunday)
Answer: #datetimezone(2011, 5, 21, 23, 59, 59.9999999, -7, 0)

Date.EndOfYear

Returns a DateTime value for the end of the year.
Syntax: Date.EndOfYear(dateTime as any) as any
Example: Date.EndOfYear(#datetime(2011, 5, 14, 17, 0, 0))
Answer: #datetime(2011, 12, 31, 23, 59, 59.9999999)

Date.From

Returns a date value from a value.
Syntax: Date.From(value as any, optional culture as nullable text) as nullable date
Example: Date.From(43910)
Answer: #date(2020, 3, 20)
Example: Date.From(#datetime(1899, 12, 30, 06, 45, 12))
Answer: #date(1899, 12, 30)

Date.Month

Excel equivalent: MONTH
Returns the month from a DateTime value.
Syntax: Date.Month(dateTime as any) as nullable number
Example: Date.Month(#datetime(2011, 12, 31, 9, 15, 36))

Date.MonthName

Excel equivalent: TEXT
Returns the name of the month component.
Syntax: Date.MonthName(date as any, optional culture as nullable text) as nullable text
Example: Date.MonthName(#datetime(2011, 12, 31, 5, 0, 0), "en-US")

Date.QuarterOfYear

Returns a number between 1 and 4 for the quarter of the year from a DateTime value.
Syntax: Date.QuarterOfYear(dateTime as any) as nullable number
Example: Date.QuarterOfYear(#date(2011, 12, 31))

Date.StartOfDay

Excel equivalent: INT
Returns a DateTime value for the start of the day.
Syntax: Date.StartOfDay(dateTime as any) as any
Example: Date.StartOfDay(#datetime(2011, 10, 10, 8, 0, 0))

Date.StartOfMonth

Excel equivalent: DATE
Returns a DateTime value representing the start of the month.
Syntax: Date.StartOfMonth(dateTime as any) as any
Example: Date.StartOfMonth(#datetime(2011, 10, 10, 8, 10, 32))

Date.StartOfQuarter

Excel equivalent: DATE
Returns a DateTime value representing the start of the quarter.
Syntax: Date.StartOfQuarter(dateTime as any) as any
Example: Date.StartOfQuarter(#datetime(2011, 10, 10, 8, 0, 0))

Date.StartOfWeek

Excel equivalent: WEEKDAY

Returns a DateTime value representing the start of the week.
Syntax: Date.StartOfWeek(dateTime as any, optional firstDayOfWeek as nullable number) as any
Example: Date.StartOfWeek(#datetime(2011, 10, 10, 8, 10, 32))

Date.StartOfYear

Returns a DateTime value representing the start of the year.
Syntax: Date.StartOfYear(dateTime as any) as any
Example: Date.StartOfYear(#datetime(2011, 10, 10, 8, 10, 32))

Date.WeekOfMonth

Returns a number for the count of week in the current month.
Syntax: Date.WeekOfMonth(dateTime as any, optional firstDayOfWeek as nullable number) as nullable number
Example: Date.WeekOfMonth(#date(2011, 03, 15))

Date.WeekOfYear

Excel equivalent: WEEKNUM
Returns a number for the count of week in the current year.
Syntax: Date.WeekOfYear(dateTime as any, optional firstDayOfWeek as nullable number) as nullable number
Example: Date.WeekOfYear(#date(2011, 03, 27))

Date.Year

Excel equivalent: YEAR
Returns the year from a DateTime value.
Syntax: Date.Year(dateTime as any) as nullable number
Example: Date.Year(#datetime(2011, 12, 31, 9, 15, 36))

DateTime.Date

Returns a date part from a DateTime value
Syntax: DateTime.Date(dateTime as any) as nullable date
Example: DateTime.Date(#datetime(2010, 12, 31, 11, 56, 02))
Answer: #date(2010, 12, 31)

DateTime.FixedLocalNow

Returns a DateTime value set to the current date and time on the system.
Syntax: DateTime.FixedLocalNow() as datetime

DateTime.LocalNow

Returns a datetime value set to the current date and time on the system.
Syntax: DateTime.LocalNow() as datetime

DateTime.Time

Returns a time part from a DateTime value.
Syntax: DateTime.Time(dateTime as any) as nullable time
Example: DateTime.Time(#datetime(2010, 12, 31, 11, 56, 02))
Answer: #time(11, 56, 2)

DateTimeZone.FixedLocalNow

Returns a DateTimeZone value set to the current date, time, and timezone offset on the system.
Syntax: DateTimeZone.FixedLocalNow() as datetimezone

DateTimeZone.FixedUtcNow

Returns the current date and time in UTC (the GMT timezone).
Syntax: DateTimeZone.FixedUtcNow() as datetimezone

DateTimeZone.LocalNow

Returns a DateTime value set to the current system date and time.

Syntax: DateTimeZone.LocalNow() as datetimezone

Duration.Days

Returns the day component of a Duration value.

Syntax: Duration.Days(duration as nullable duration) as nullable number

Example: Duration.Days(#duration(5, 4, 3, 2))

Answer: 5

Duration.Hours

Returns an hour component of a Duration value.

Syntax: Duration.Hours(duration as nullable duration) as nullable number

Example: Duration.Hours(#duration(5, 4, 3, 2))

Duration.Minutes

Returns a minute component of a Duration value.

Syntax: Duration.Minutes(duration as nullable duration) as nullable number

Example: Duration.Minutes(#duration(5, 4, 3, 2))

Duration.Seconds

Returns a second component of a Duration value.

Syntax: Duration.Seconds(duration as nullable duration) as nullable number

Example: Duration.Seconds(#duration(5, 4, 3, 2))

Duration.TotalDays

Returns the total magnitude of days from a Duration value.

Syntax: Duration.TotalDays(duration as nullable duration) as nullable number

Example: Duration.TotalDays(#duration(5, 4, 3, 2))

Answer: 5.1687731481481478

Duration.TotalHours

Returns the total magnitude of hours from a Duration value.

Syntax: Duration.TotalHours(duration as nullable duration) as nullable number

Example: Duration.TotalHours(#duration(5, 4, 3, 2))

Answer: 124.05055555555555

Duration.TotalMinutes

Returns the total magnitude of minutes from a Duration value.

Syntax: Duration.TotalMinutes(duration as nullable duration) as nullable number

Example: Duration.TotalMinutes(#duration(5, 4, 3, 2))

Answer: 7443.0333333333338

Duration.TotalSeconds

Returns the total magnitude of seconds from a duration value.

Syntax: Duration.TotalSeconds(duration as nullable duration) as nullable number

Example: Duration.TotalSeconds(#duration(5, 4, 3, 2))

Answer: 446582

Duration.ToText

Returns a text value from a Duration value.

Syntax: Duration.ToText(duration as nullable duration, optional format as nullable text) as nullable text

Example: Duration.ToText(#duration(2, 5, 55, 20))

Answer: "2.05:55:20"

Json.Document

Returns the contents of a JSON document. The contents may be directly passed to the function as text, or it may be the binary value returned by a function like File.Contents.
Syntax: Json.Document(jsonText as any, optional encoding as nullable number) as any

List.Average

Excel equivalent: AVERAGE
Returns an average value from a list in the datatype of the values in the list.
Syntax: List.Average(list as list, optional precision as nullable number) as any
Example: List.Average({3, 4, 6})
Answer: 4.333333333333333
Example: List.Average({#date(2011, 1, 1), #date(2011, 1, 2), #date(2011, 1, 3)})
Answer: #date(2011, 1, 2)

List.Count

Excel equivalent: COUNT
Returns the number of items in a list.
Syntax: List.Count(list as list) as number
Example: List.Count({1, 2, 3})
Answer: 3

List.Distinct

Filters a list down by removing duplicates. An optional equation criteria value can be specified to control equality comparison. The first value from each equality group is chosen.
Syntax: List.Distinct(list as list, optional equationCriteria as any) as list
Example: List.Distinct({1, 1, 2, 3, 3, 3})

List.FirstN

Returns the first set of items in the list by specifying how many items to return or a qualifying condition provided by countOrCondition.
Syntax: List.FirstN(list as list, countOrCondition as any) as any
Example: List.FirstN({3, 4, 5, -1, 7, 8, 2},each _ > 0)
Answer: {3, 4, 5}

List.LastN

Returns the last set of items in a list by specifying how many items to return or a qualifying condition.
Syntax: List.LastN(list as list, optional countOrCondition as any) as any
Example: List.LastN({3, 4, 5, -1, 7, 8, 2},1)
Answer: 2
Example: List.LastN({3, 4, 5, -1, 7, 8, 2}, each _ > 0)
Answer: {7, 8, 2}

List.Max

Excel equivalent: MAX
Returns the maximum item in a list, or the optional default value if the list is empty.
Syntax: List.Max(list as list, optional default as any, optional comparisonCriteria as any, optional includeNulls as nullable logical) as any
Example: List.Max({1, 4, 7, 3, -2, 5},1)
Answer: 7
Example: List.Max({}, -1)
Answer: -1

List.Median

Excel equivalent: MEDIAN

Returns the median item from a list.
Syntax: List.Median(list as list, optional comparisonCriteria as any) as any
Example: powerquery-mList.Median({5, 3, 1, 7, 9})
Answer: 5

List.Min

Excel equivalent: MIN
Returns the minimum item in a list, or the optional default value if the list is empty.
Syntax: List.Min(list as list, optional default as any, optional comparisonCriteria as any, optional includeNulls as nullable logical) as any
Example: List.Min({1, 4, 7, 3, -2, 5})
Answer: -2
Example: List.Min({}, -1)
Answer: -1

List.NonNullCount

Excel equivalent: COUNTA
Returns the number of items in a list excluding null values
Syntax: List.NonNullCount(list as list) as number

List.Sum

Excel equivalent: SUM
Returns the sum from a list.
Syntax: List.Sum(list as list, optional precision as nullable number) as any
Example: List.Sum({1, 2, 3})
Answer: 6

Number.Abs

Excel equivalent: ABS
Returns the absolute value of a number.
Syntax: Number.Abs(number as nullable number) as nullable number
Example: Number.Abs(-9)

Number.IntegerDivide

Divides two numbers and returns the whole part of the resulting number.
Syntax: Number.IntegerDivide(number1 as nullable number, number2 as nullable number, optional precision as nullable number) as nullable number
Example: Number.IntegerDivide(8.3, 3) = 2

Number.Mod

Excel equivalent: MOD
Divides two numbers and returns the remainder of the resulting number.
Syntax: Number.Mod(number as nullable number, divisor as nullable number, optional precision as nullable number) as nullable number
Example: Number.Mod(5, 3)

Number.Power

Excel equivalent: POWER
Returns a number raised by a power.
Syntax: Number.Power(number as nullable number, power as nullable number) as nullable number
Example: Number.Power(5, 3)

Number.Round

Excel equivalent: ROUND

Returns a nullable number (n) if value is an integer.
Syntax: Number.Round(number as nullable number, optional digits as nullable number [0], optional roundingMode as nullable number) as nullable number
Example: Number.Round(1.2345, 3, RoundingMode.Up) Number.Round(1.2345, 3, RoundingMode.Down)

Number.RoundAwayFromZero

Excel equivalent: ROUNDUP
Returns Number.RoundUp(value) when value >= 0 and Number.RoundDown(value) when value < 0.
Syntax: Number.RoundAwayFromZero(number as nullable number [0], optional digits as nullable number) as nullable number
Example: Number.RoundAwayFromZero(-1.234, 2)

Number.RoundDown

Excel equivalent: FLOOR
Returns the largest integer less than or equal to a number value.
Syntax: Number.RoundDown(number as nullable number, optional digits as nullable number) as nullable number
Example: Number.RoundDown(1.999, 2)
Answer: 1.99

Number.RoundTowardZero

Excel equivalent: ROUNDDOWN
Returns Number.RoundDown(x) when x >= 0 and Number.RoundUp(x) when x < 0.
Syntax: Number.RoundTowardZero(number as nullable number, optional digits as nullable number) as nullable number

Number.RoundUp

Excel equivalent: CEILING
Returns the larger integer greater than or equal to a number value.
Syntax: Number.RoundUp(number as nullable number, optional digits as nullable number) as nullable number
Example: Number.RoundUp(1.234)
Example: Number.RoundUp(1.999)
Example: Number.RoundUp(1.234, 2)

Number.Sign

Excel equivalent: SIGN
Returns 1 for positive numbers, -1 for negative numbers or 0 for zero.
Syntax: Number.Sign(number as nullable number) as nullable number
Example: Number.Sign(182)
Example: Number.Sign(-182)
Example: Number.Sign(0)

Number.Sqrt

Excel equivalent: SQRT
Returns the square root of a number.
Syntax: Number.Sqrt(number as nullable number) as nullable number
Example: Number.Sqrt(625)
Example: Number.Sqrt(85)
Answer: 9.21954445729288

Table.AddColumn

Adds a column named newColumnName to a table.
Syntax: Table.AddColumn(table as table, newColumnName as text, columnGenerator as function, optional

Example: Table.AddColumn(Table.FromRecords({[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0, Shipping

Answer: = 10.00], [OrderID = 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0, Shipping = 15.00], [OrderID = 3,

Table.AlternateRows

Returns a table containing an alternating pattern of the rows from a table.

Syntax: Table.AlternateRows(table as table, offset as number, skip as number, take as number) as table

Example: Table.AlternateRows(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2,

Answer: Name = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"]}), 1, 1, 1)

Table.Combine

Returns a table that is the result of merging a list of tables. The tables must all have the same row type structure.

Syntax: Table.Combine(tables as list, optional columns as any) as table

Example: Table.Combine({Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"]}),

Answer: Table.FromRecords({[CustomerID = 2, Name = "Jim", Phone = "987-6543"] }),Table.FromRecords({[CustomerID = 3,

Example: Table.Combine({Table.FromRecords({[Name="Bob",Phone="123-4567"]}), Table.FromRecords({[Fax="987-6543",

Answer: Phone="838-7171"] }),Table.FromRecords({[Cell = "543-7890"]})})

Example: Phone="838-7171"] }),Table.FromRecords({[Cell = "543-7890"]})}, {"CustomerID", "Name"})

Answer: CUSTOMERID NAME

Table.DemoteHeaders

Demotes the header row down into the first row of a table.

Syntax: Table.DemoteHeaders(table as table) as table

Example: Table.DemoteHeaders(Table.FromRecords({[CustomerID=1, Name="Bob", Phone="123-4567"],[CustomerID=2, Name="Jim",

Answer: Phone="987-6543"]}))

Table.ExpandTableColumn

Expands a column of records or a column of tables into multiple columns in the containing table.

Syntax: Table.ExpandTableColumn(table as table, column as text, columnNames as list, optional

Example: [t.a] , [t.b] and [t.c] .

Answer: Table.ExpandTableColumn(Table.FromRecords({[t = Table.FromRecords({[a=1, b=2, c= 3],[a=2,b=4,c=6]}), b = 2]}),

Table.FillDown

Replaces null values in the specified column or columns of the table with the most recent non-null value in the column.

Syntax: Table.FillDown(table as table, columns as list) as table

Example: Table.FillDown(Table.FromRecords({[Place=1, Name="Bob"], [Place=null, Name="John"], [Place=2, Name="Brad"],

Answer: [Place=3, Name="Mark"], [Place=null, Name="Tom"], [Place=null, Name="Adam"]}), {"Place"})

Table.FillUp

Returns a table from the table specified where the value of the next cell is propagated to the null values cells above in the column specified.

Syntax: Table.FillUp(table as table, columns as list) as table

Example: Table.FillUp(Table.FromRecords({[Column1 = 1, Column2 = 2], [Column1 = 3, Column2 = null], [Column1 = 5,

Answer: Column2 = 3]}), {"Column2"})

Table.FirstN

Returns the first row(s) of a table, depending on the countOrCondition parameter.

Syntax: Table.FirstN(table as table, countOrCondition as any) as table

Example: Table.FirstN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name =

Answer: "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"]}), 2)

Example: Table.FirstN(Table.FromRecords({[a = 1, b = 2], [a = 3, b = 4], [a = -5, b = -6]}), each [a] > 0)

Answer: A B

Table.Group

Groups table rows by the values of key columns for each row.

Syntax: Table.Group(table as table, key as any, aggregatedColumns as list, optional groupKind as nullable

Example: Table.Group(Table.FromRecords({[CustomerID= 1, price = 20], [CustomerID= 2, price = 10], [CustomerID= 2, price

Answer: = 20], [CustomerID= 1, price = 10], [CustomerID= 3, price = 20], [CustomerID= 3, price = 5]}), "CustomerID",

Table.LastN

Returns the last row(s) from a table, depending on the countOrCondition parameter.

Syntax: Table.LastN(table as table, countOrCondition as any) as table

Example: Table.LastN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name =

Answer: "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"]}), 2)

Example: Table.LastN(Table.FromRecords({[a = -1, b = -2], [a = 3, b = 4], [a = 5, b = 6]}), each _ [a] > 0)

Answer: A B

Table.Max

Returns the largest row or rows from a table using a comparisonCriteria.

Syntax: Table.Max(table as table, comparisonCriteria as any, optional default as any) as any

Example: Table.Max(Table.FromRecords({[a = 2, b = 4], [a = 6, b = 8]}), "a")

Answer: A 6

Example: Table.Max(#table({"a"},{}), "a", -1)

Answer: -1

Table.Min

Returns the smallest row or rows from a table using a comparisonCriteria.

Syntax: Table.Min(table as table, comparisonCriteria as any, optional default as any) as any

Example: Table.Min(Table.FromRecords({[a = 2, b = 4], [a = 6, b = 8]}), "a")

Answer: A 2

Example: Table.Min(#table({"a"},{}), "a", -1)

Table.NestedJoin

Joins the rows of the tables based on the equality of the keys. The results are entered into a new column.

Syntax: Table.NestedJoin(table1 as table, key1 as any, table2 as any, key2 as any, newColumnName as text,

Table.Pivot

Given a table and attribute column containing pivotValues, creates new columns for each of the pivot values and assigns them values from the valueColumn. An optional aggregationFunction can be provided to handle multiple occurrence of the same key value in the attribute column.

Syntax: Table.Pivot(table as table, pivotValues as list, attributeColumn as text, valueColumn as text,

Example: = "a", value = 2], [key = "y", attribute = "b", value = 4] })

Answer: and pivot them into their own column.

Example: = "c", value = 5], [key = "y", attribute = "a", value = 2], [key = "y", attribute = "b", value = 4] })

Answer: and pivot them into their own column. The attribute "c" for key "x" has multiple values associated with it, so use the

Table.PromoteHeaders

Promotes the first row of the table into its header or column names.
Syntax: Table.PromoteHeaders(table as table, optional options as nullable record) as table
Example: Table.PromoteHeaders(Table.FromRecords({[Column1 = "CustomerID", Column2 = "Name", Column3 = #date(1980,1,1)],
Answer: [Column1 = 1, Column2 = "Bob", Column3 = #date(1980,1,1)]}))
Example: Table.PromoteHeaders(Table.FromRecords({[Rank = 1, Name = "Name", Date = #date(1980,1,1)],[Rank = 1, Name =
Answer: "Bob", Date = #date(1980,1,1)]}), [PromoteAllScalars = true, Culture = "en-US"])

Table.Range

Returns the specified number of rows from a table starting at an offset.
Syntax: Table.Range(table as table, offset as number, optional count as nullable number) as table
Example: Table.Range(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name =
Answer: "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =
Example: Table.Range(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name =
Answer: "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Table.RemoveColumns

Returns a table without a specific column or columns.
Syntax: Table.RemoveColumns(table as table, columns as any, optional missingField as nullable number) as
Example: Table.RemoveColumns(Table.FromRecords({[CustomerID=1, Name="Bob", Phone = "123-4567"]}), "Phone")
Answer: CUSTOMERID NAME
Example: Table.RemoveColumns(Table.FromRecords({[CustomerID=1, Name="Bob", Phone = "123-4567"]}), "Address")
Answer: [Expression.Error] The field 'Address' of the record was not found.

Table.RemoveFirstN

Returns a table with the specified number of rows removed from the table starting at the first row. The number of rows removed depends on the optional countOrCondition parameter.
Syntax: Table.RemoveFirstN(table as table, countOrCondition as any) as table
Example: Table.RemoveFirstN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2,
Answer: Name = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name
Example: Table.RemoveFirstN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2,
Answer: Name = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name
Example: Table.RemoveFirstN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2,
Answer: Name = "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] , [CustomerID = 4,

Table.RemoveLastN

Returns a table with the specified number of rows removed from the table starting at the last row. The number of rows removed depends on the optional countOrCondition parameter.

Syntax: Table.RemoveLastN(table as table, optional countOrCondition as any) as table

Example: Table.RemoveLastN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"],[CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"],[CustomerID = 3, Name = "Paul", Phone = "543-7890"],[CustomerID = 4, Name =

Example: Table.RemoveLastN(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"],[CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"],[CustomerID = 3, Name = "Paul", Phone = "543-7890"],[CustomerID = 4, Name =

Table.RemoveRows

Returns a table with the specified number of rows removed from the table starting at an offset.

Syntax: Table.RemoveRows(table as table, offset as number, optional count as nullable number) as table

Example: Table.RemoveRows(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Example: Table.RemoveRows(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Example: Table.RemoveRows(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Table.RemoveRowsWithErrors

Returns a table with all rows removed from the table that contain an error in at least one of the cells in a row.

Syntax: Table.RemoveRowsWithErrors(table as table, optional columns as nullable list) as table

Example: Table.RemoveRowsWithErrors(Table.FromRecords({[Column1=...],[Column1=2], [Column1=3]}))

Table.RenameColumns

Returns a table with the columns renamed as specified.

Syntax: Table.RenameColumns(table as table, renames as list, optional missingField as nullable number) as

Example: Table.RenameColumns(Table.FromRecords({[CustomerNum=1, Name="Bob", Phone = "123-4567"]}), {"CustomerNum",

Answer: "CustomerID"})

Example: Table.RenameColumns(Table.FromRecords({[CustomerNum=1, Name="Bob", PhoneNum = "123-4567"]}), {{"CustomerNum",

Answer: "CustomerID"}, {"PhoneNum", "Phone"}})

Example: Table.RenameColumns(Table.FromRecords({[CustomerID=1, Name="Bob", Phone = "123-4567"]}), {"NewCol",

Answer: "NewColumn"}, MissingField.Ignore)

Table.ReorderColumns

Returns a table with specific columns in an order relative to one another.

Syntax: Table.ReorderColumns(table as table, columnOrder as list, optional missingField as nullable

Example: Table.ReorderColumns(Table.FromRecords({[CustomerID=1, Phone = "123-4567", Name ="Bob"]}), {"Name","Phone"})

Answer: CUSTOMERID NAME PHONE

Example: Table.ReorderColumns(Table.FromRecords({[CustomerID=1, Name = "Bob", Phone = "123-4567"]}), {"Phone",

Answer: "Address"}, MissingField.Ignore)

Table.ReplaceErrorValues

Replaces the error values in the specified columns with the corresponding specified value.

Syntax: Table.ReplaceErrorValues(table as table, errorReplacement as list) as table

Example: Table.ReplaceErrorValues(Table.FromRows({{1,"hello"},{3,...}}, {"A","B"}), {"B", "world"})

Answer: A B

Example: Table.ReplaceErrorValues(Table.FromRows({{..., ...},{1,2}}, {"A","B"}), {{"A", "hello"}, {"B", "world"}})

Answer: A B

Table.ReplaceValue

Replaces oldValue with newValue in specific columns of a table, using the provided replacer function, such as text.Replace or Value.Replace.

Syntax: Table.ReplaceValue(table as table, oldValue as any, newValue as any, replacer as function,

Example: Table.ReplaceValue(Table.FromRecords({[a = 1, b = "hello"], [a = 3, b = "goodbye"]}), "goodbye", "world",

Answer: Replacer.ReplaceText, {"b"})

Example: Table.ReplaceValue(Table.FromRecords({[a = 1, b = "hello"], [a = 3, b = "wurld"]}), "ur", "or",

Answer: Replacer.ReplaceText, {"b"})

Table.ReverseRows

Returns a table with the rows in reverse order.

Syntax: Table.ReverseRows(table as table) as table

Example: Table.ReverseRows(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Table.RowCount

Returns the number of rows in a table.

Syntax: Table.RowCount(table as table) as number

Example: Table.RowCount(Table.FromRecords({[CustomerID =1, Name ="Bob", Phone = "123-4567"],[CustomerID =2, Name

Answer: ="Jim", Phone = "987-6543"],[CustomerID =3, Name ="Paul", Phone = "543-7890"]}))

Table.SelectColumns

Returns a table that contains only specific columns.

Syntax: Table.SelectColumns(table as table, columns as any, optional missingField as nullable number) as

Example: Table.SelectColumns(Table.FromRecords({ [CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2,

Answer: Name = "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] , [CustomerID = 4,

Example: Table.SelectColumns(Table.FromRecords({[CustomerID=1, Name="Bob", Phone = "123-4567"]}), {"CustomerID",

Answer: "Name"})

Example: Table.SelectColumns(Table.FromRecords({[CustomerID=1, Name="Bob", Phone = "123-4567"]}), "NewColumn")

Answer: [Expression.Error] The field 'NewColumn' of the record wasn't found.

Example: Table.SelectColumns(Table.FromRecords({[CustomerID=1, Name = "Bob", Phone = "123-4567"]}), {"CustomerID",

Answer: "NewColumn"}, MissingField.UseNull)

Table.SelectRows

Returns a table containing only the rows that match a condition.

Syntax: Table.SelectRows(table as table, condition as function) as table

Example: Table.SelectRows(Table.FromRecords({ [CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] , [CustomerID = 4, Name =

Example: Table.SelectRows(Table.FromRecords({ [CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name

Answer: = "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] , [CustomerID = 4, Name =

Table.SelectRowsWithErrors

Returns a table with only the rows from table that contain an error in at least one of the cells in a row.

Syntax: Table.SelectRowsWithErrors(table as table, optional columns as nullable list) as table

Example: Table.SelectRowsWithErrors(Table.FromRecords({ [CustomerID =..., Name = "Bob", Phone = "123-4567"],

Answer: [CustomerID = 2, Name = "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] ,

Table.Skip

Returns a table that does not contain the first row or rows of the table.

Syntax: Table.Skip(table as table, countOrCondition as any) as table

Example: Table.Skip(Table.FromRecords({ [CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name =

Answer: "Jim", Phone = "987-6543"] , [CustomerID = 3, Name = "Paul", Phone = "543-7890"] , [CustomerID = 4, Name =

Example: Table.Skip(Table.FromRecords({[CustomerID = 1, Name = "Bob", Phone = "123-4567"],[CustomerID = 2, Name =

Answer: "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Example: Table.Skip(Table.FromRecords({[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0], [OrderID =

Answer: 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0], [OrderID = 3, CustomerID = 2, Item = "Fishing net",

Table.Sort

Sorts the rows in a table using a comparisonCriteria or a default ordering if one is not specified.

Syntax: Table.Sort(table as table, comparisonCriteria as any) as table

Example: Table.Sort(Table.FromRecords({[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0], [OrderID =

Answer: 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0], [OrderID = 3, CustomerID = 2, Item = "Fishing net",

Example: Table.Sort(Table.FromRecords({[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0], [OrderID =

Answer: 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0], [OrderID = 3, CustomerID = 2, Item = "Fishing net",

Example: Table.Sort(Table.FromRecords({[OrderID = 1, CustomerID = 1, Item = "Fishing rod", Price = 100.0], [OrderID =

Answer: 2, CustomerID = 1, Item = "1 lb. worms", Price = 5.0], [OrderID = 3, CustomerID = 2, Item = "Fishing net",

Table.SplitColumn

Returns a new set of columns from a single column applying a splitter function to each value.

Syntax: Table.SplitColumn(table as table, sourceColumn as text, splitter as function, optional
Example: let Customers = Table.FromRecords({ [CustomerID = 1, Name = "Bob", Phone = "123-4567"], [CustomerID = 2, Name
Answer: = "Jim", Phone = "987-6543"], [CustomerID = 3, Name = "Paul", Phone = "543-7890"], [CustomerID = 4, Name =

Table.TransformColumns

Transforms columns from a table using a function.
Syntax: Table.TransformColumns(table as table, transformOperations as list, optional defaultTransformation
Example: Table.TransformColumns(Table.FromRecords({[A="1", B=2], [A="5", B=10]}),{"A", Number.FromText})
Answer: A B
Example: Table.TransformColumns(Table.FromRecords({[A="1", B=2], [A="5", B=10]}), {"X", Number.FromText}, null,
Answer: MissingField.Ignore)
Example: Table.TransformColumns(Table.FromRecords({[A="1",B=2], [A="5", B=10]}), {"X", Number.FromText}, null,
Answer: MissingField.UseNull)
Example: Table.TransformColumns(Table.FromRecords({[A="1",B=2], [A="5", B=10]}), {"X", Number.FromText})
Answer: [Expression.Error] The column 'X' of the table wasn't found.

Table.TransformColumnTypes

Transforms the column types from a table using a type.
Syntax: Table.TransformColumnTypes(table as table, typeTransformations as list, optional culture as
Example: Table.TransformColumnTypes(Table.FromRecords({[a = 1, b = 2], [a = 3, b = 4]}), {"a", type text}, "en-US")
Answer: A B

Table.Transpose

Returns a table with columns converted to rows and rows converted to columns from the input table.
Syntax: Table.Transpose(table as table, optional columns as any) as table
Example: Table.Transpose(Table.FromRecords({[Name = "Full Name", Value = "Fred"], [Name = "Age", Value = 42], [Name =
Answer: "Country", Value = "UK"]}))

Table.Unpivot

Given a list of table columns, transforms those columns into attribute-value pairs.
Syntax: Table.Unpivot(table as table, pivotColumns as list, attributeColumn as text, valueColumn as text)
Example: ({[key = "x", a = 1, b = null, c = 3], [key = "y", a = 2, b = 4, c = null]})
Answer: attribute-value pairs.

Table.UnpivotOtherColumns

Translates all columns other than a specified set into attribute-value pairs, combined with the rest of the values in each row.
Syntax: Table.UnpivotOtherColumns(table as table, pivotColumns as list, attributeColumn as text,
Example: Table.UnpivotOtherColumns(Table.FromRecords({ [key = "key1", attribute1 = 1, attribute2 = 2, attribute3 = 3
Answer:], [key = "key2", attribute1 = 4, attribute2 = 5, attribute3 = 6] }), { "key" }, "column1", "column2")

Text.AfterDelimiter

Returns the portion of text after the specified delimiter.
Syntax: Text.AfterDelimiter(text as nullable text, delimiter as text, optional index as any) as any
Example: Text.AfterDelimiter("111-222-333", "-")
Example: Text.AfterDelimiter("111-222-333", "-", 1)
Example: Text.AfterDelimiter("111-222-333", "-", {1, RelativePosition.FromEnd})

Text.BeforeDelimiter

Returns the portion of text before the specified delimiter.

Syntax: Text.BeforeDelimiter(text as nullable text, delimiter as text, optional index as any) as any

Example: Text.BeforeDelimiter("111-222-333", "-")

Example: Text.BeforeDelimiter("111-222-333", "-", 1)

Example: Text.BeforeDelimiter("111-222-333", "-", {1, RelativePosition.FromEnd})

Text.BetweenDelimiters

Returns the portion of text between the specified startDelimiter and endDelimiter.

Syntax: Text.BetweenDelimiters(text as nullable text, startDelimiter as text, endDelimiter as text,

Example: Text.BetweenDelimiters("111 (222) 333 (444)", "(", ")")

Answer: 222

Example: Text.BetweenDelimiters("111 (222) 333 (444)", "(", ")", 1, 0)

Answer: 444

Example: Text.BetweenDelimiters("111 (222) 333 (444)", "(", ")", {1, RelativePosition.FromEnd}, {1,

Answer: RelativePosition.FromStart})

Text.Clean

Returns the original text value with non-printable characters removed.

Syntax: Text.Clean(text as nullable text) as nullable text

Example: Text.Clean("ABC#(lf)D")

Text.Combine

Returns a text value that is the result of joining all text values with each value separated by a separator.

Syntax: Text.Combine(texts as list, optional separator as nullable text) as text

Example: Text.Combine({"Seattle", "WA"})

Example: Text.Combine({"Seattle", "WA"}, ", ")

Answer: Seattle, WA

Text.End

Returns the number of characters from the end of a text value.

Syntax: Text.End(text as nullable text, count as number) as nullable text

Example: Text.End("Hello, World", 5)

Text.Length

Returns the number of characters in a text value.

Syntax: Text.Length(text as nullable text) as nullable number

Example: Text.Length("Hello World")

Text.Lower

Returns the lowercase of a text value.

Syntax: Text.Lower(text as nullable text, optional culture as nullable text) as nullable text

Example: Text.Lower("AbCd")

Text.Middle

Returns the substring up to a specific length.

Syntax: Text.Middle(text as nullable text, start as number, optional count as nullable number) as nullable

Example: Text.Middle("Hello World", 6, 5)

Example: Text.Middle("Hello World", 6, 20)

Text.Proper

Returns a text value with first letters of all words converted to uppercase.

Syntax: Text.Proper(text as nullable text, optional culture as nullable text) as nullable text

Example: Text.Proper

Answer: Text.Proper("the QUICK BrOWn fOx jUmPs oVER tHe LAzy DoG")

Text.Range

Returns a number of characters from a text value starting at a zero-based offset and for count number of characters.

Syntax: Text.Range(text as nullable text, offset as number, optional count as nullable number) as nullable

Example: Text.Range("Hello World", 6)

Example: Text.Range("Hello World Hello", 6, 5)

Text.Start

Returns the count of characters from the start of a text value.

Syntax: Text.Start(text as nullable text, count as number) as nullable text

Example: Text.Start("Hello, World", 5)

Text.Trim

Removes any occurrences of characters in trimChars from text.

Syntax: Text.Trim(text as nullable text, optional trim as any) as nullable text

Example: Text.Trim(" a b c d ")

Text.TrimEnd

Removes any occurrences of the characters specified in trimChars from the end of the original text value.

Syntax: Text.TrimEnd(text as nullable text, optional trim as any) as nullable text

Example: Text.TrimEnd(" a b c d ")

Text.TrimStart

Removes any occurrences of the characters in trimChars from the start of the original text value.

Syntax: Text.TrimStart(text as nullable text, optional trim as any) as nullable text

Example: Text.TrimStart(" a b c d ")

Text.Upper

Returns the uppercase of a text value.

Syntax: Text.Upper(text as nullable text, optional culture as nullable text) as nullable text

Example: Text.Upper("aBcD")

Time.EndOfHour

Returns a DateTime value from the end of the hour.

Syntax: Time.EndOfHour(dateTime as any) as any

Example: Time.EndOfHour(#datetime(2011, 5, 14, 17, 0, 0))

Answer: #datetime(2011, 5, 14, 17, 59, 59.9999999)

Example: Time.EndOfHour(#datetimezone(2011, 5, 17, 5, 0, 0, -7, 0))

Answer: #datetimezone(2011, 5, 17, 5, 59, 59.9999999, -7, 0)

Time.Hour

Returns an hour value from a DateTime value.

Syntax: Time.Hour(dateTime as any) as nullable number

Example: Time.Hour(#datetime(2011, 12, 31, 9, 15, 36))

Time.Minute

Returns a minute value from a DateTime value.

Syntax: Time.Minute(dateTime as any) as nullable number

Example: Time.Minute(#datetime(2011, 12, 31, 9, 15, 36))

Time.Second

Returns a second value from a DateTime value

Syntax: Time.Second(dateTime as any) as nullable number`

Example: Time.Second(#datetime(2011, 12, 31, 9, 15, 36.5))

Time.StartOfHour

Returns the first value of the hour from a time value.

Syntax: Time.StartOfHour(dateTime as any) as any

Example: Time.StartOfHour(#datetime(2011, 10, 10, 8, 10, 32))

Answer: #datetime(2011, 10, 10, 8, 0, 0)

Time.ToRecord

Returns a record containing parts of a Date value.

Syntax: Time.ToRecord(time as time) as record

Example: Time.ToRecord(#time(11, 56, 2))

Answer: HOUR 11

Time.ToText

Returns a text value from a Time value.

Syntax: Time.ToText(time as nullable time, optional format as nullable text, optional culture as nullable

Example: Time.ToText(#time(11, 56, 2))

Answer: 0.497222222222222

Example: Time.ToText(#time(11, 56, 2), "hh:mm")

Answer: 0.497222222222222