

Language Index

This section contains an index to all of the functions available in Wrangle .

Aggregate Functions

Item	Description
<i>ANY Function</i>	Extracts a non-null and non-missing value from a specified column. If all values are missing or null, the function returns a null value. See <i>ANY Function</i> .
<i>ANYIF Function</i>	Selects a single non-null value from rows in each group that meet a specific condition. See <i>ANYIF Function</i> .
<i>APPROXIMATEMEDIAN Function</i>	Computes the approximate median from all row values in a column or group. Input column can be of Integer or Decimal. See <i>APPROXIMATEMEDIAN Function</i> .
<i>APPROXIMATEPERCENTILE Function</i>	Computes an approximation for a specified percentile across all row values in a column or group. Input column can be of Integer or Decimal. See <i>APPROXIMATEPERCENTILE Function</i> .
<i>APPROXIMATEQUARTILE Function</i>	Computes an approximation for a specified quartile across all row values in a column or group. Input column can be of Integer or Decimal. See <i>APPROXIMATEQUARTILE Function</i> .
<i>AVERAGE Function</i>	Computes the average (mean) from all row values in a column or group. Input column can be of Integer or Decimal. See <i>AVERAGE Function</i> .
<i>AVERAGEIF Function</i>	Generates the average value of rows in each group that meet a specific condition. Generated value is of Decimal type. See <i>AVERAGEIF Function</i> .
<i>CORREL Function</i>	Computes the correlation coefficient between two columns. Source values can be of Integer or Decimal type. See <i>CORREL Function</i> .
<i>COUNTA Function</i>	Generates the count of non-null rows in a specified column, optionally counted by group. Generated value is of Integer type. See <i>COUNTA Function</i> .
<i>COUNTAIF Function</i>	Generates the count of non-null values for rows in each group that meet a specific condition. See <i>COUNTAIF Function</i> .
<i>COUNTDISTINCT Function</i>	Generates the count of distinct values in a specified column, optionally counted by group. Generated value is of Integer type. See <i>COUNTDISTINCT Function</i> .
<i>COUNTDISTINCTIF Function</i>	Generates the count of distinct non-null values for rows in each group that meet a specific condition. See <i>COUNTDISTINCTIF Function</i> .
<i>COUNT Function</i>	Generates the count of rows in the dataset. Generated value is of Integer type. See <i>COUNT Function</i> .
<i>COUNTIF Function</i>	Generates the count of rows in each group that meet a specific condition. Generated value is of Integer type. See <i>COUNTIF Function</i> .
<i>COVAR Function</i>	Computes the covariance between two columns using the population method. Source values can be of Integer or Decimal type. See <i>COVAR Function</i> .
<i>COVARSAMP Function</i>	Computes the covariance between two columns using the sample method. Source values can be of Integer or Decimal type. See <i>COVARSAMP Function</i> .
<i>KTHLARGEST Function</i>	Extracts the ranked value from the values in a column, where $k=1$ returns the maximum value. The value for k must be between 1 and 1000, inclusive. Inputs can be Integer, Decimal, or Datetime. See <i>KTHLARGEST Function</i> .
<i>KTHLARGESTIF Function</i>	Extracts the ranked value from the values in a column, where $k=1$ returns the maximum value, when a specified condition is met. The value for k must be between 1 and 1000, inclusive. Inputs can be Integer, Decimal, or Datetime. See <i>KTHLARGESTIF Function</i> .
<i>KTHLARGESTUNIQUE Function</i>	Extracts the ranked unique value from the values in a column, where $k=1$ returns the maximum value. The value for k must be between 1 and 1000, inclusive. Inputs can be Integer, Decimal, or Datetime. See <i>KTHLARGESTUNIQUE Function</i> .

<i>KTHLARGEST UNIQUEIF Function</i>	Extracts the ranked unique value from the values in a column, where $k=1$ returns the maximum value, when a specified condition is met. The value for k must be between 1 and 1000, inclusive. Inputs can be Integer, Decimal, or Datetime. See <i>KTHLARGESTUNIQUEIF Function</i> .
<i>LIST Function</i>	Extracts the set of values from a column into an array stored in a new column. This function is typically part of an aggregation. See <i>LIST Function</i> .
<i>LISTIF Function</i>	Returns list of all values in a column for rows that match a specified condition. See <i>LISTIF Function</i> .
<i>MAX Function</i>	Computes the maximum value found in all row values in a column. Inputs can be Integer, Decimal, or Datetime. See <i>MAX Function</i> .
<i>MAXIF Function</i>	Generates the maximum value of rows in each group that meet a specific condition. Inputs can be Integer, Decimal, or Datetime. See <i>MAXIF Function</i> .
<i>MEDIAN Function</i>	Computes the median from all row values in a column or group. Input column can be of Integer or Decimal. See <i>MEDIAN Function</i> .
<i>MIN Function</i>	Computes the minimum value found in all row values in a column. Input column can be of Integer, Decimal or Datetime. See <i>MIN Function</i> .
<i>MINIF Function</i>	Generates the minimum value of rows in each group that meet a specific condition. Inputs can be Integer, Decimal, or Datetime. See <i>MINIF Function</i> .
<i>MODE Function</i>	Computes the mode (most frequent value) from all row values in a column, according to their grouping. Input column can be of Integer, Decimal, or Datetime type. See <i>MODE Function</i> .
<i>MODEIF Function</i>	Computes the mode (most frequent value) from all row values in a column, according to their grouping. Input column can be of Integer, Decimal, or Datetime type. See <i>MODEIF Function</i> .
<i>PERCENTILE Function</i>	Computes a specified percentile across all row values in a column or group. Input column can be of Integer or Decimal. See <i>PERCENTILE Function</i> .
<i>QUARTILE Function</i>	Computes a specified quartile across all row values in a column or group. Input column can be of Integer or Decimal. See <i>QUARTILE Function</i> .
<i>STDEV Function</i>	Computes the standard deviation across all column values of Integer or Decimal type. See <i>STDEV Function</i> .
<i>STDEVIF Function</i>	Generates the standard deviation of values by group in a column that meet a specific condition. See <i>STDEVIF Function</i> .
<i>STDEVSAMP Function</i>	Computes the standard deviation across column values of Integer or Decimal type using the sample statistical method. See <i>STDEVSAMP Function</i> .
<i>STDEVSAMPIF Function</i>	Generates the standard deviation of values by group in a column that meet a specific condition using the sample statistical method. See <i>STDEVSAMPIF Function</i> .
<i>SUM Function</i>	Computes the sum of all values found in all row values in a column. Input column can be of Integer or Decimal. See <i>SUM Function</i> .
<i>SUMIF Function</i>	Generates the sum of rows in each group that meet a specific condition. See <i>SUMIF Function</i> .
<i>UNIQUE Function</i>	Extracts the set of unique values from a column into an array stored in a new column. This function is typically part of an aggregation. See <i>UNIQUE Function</i> .
<i>VAR Function</i>	Computes the variance among all values in a column. Input column can be of Integer or Decimal. If no numeric values are detected in the input column, the function returns 0. See <i>VAR Function</i> .
<i>VARIF Function</i>	Generates the variance of values by group in a column that meet a specific condition. See <i>VARIF Function</i> .
<i>VARSAAMP Function</i>	Computes the variance among all values in a column using the sample statistical method. Input column can be of Integer or Decimal. If no numeric values are detected in the input column, the function returns 0. See <i>VARSAAMP Function</i> .
<i>VARSAAMPIF Function</i>	Generates the variance of values by group in a column that meet a specific condition using the sample statistical method. See <i>VARSAAMPIF Function</i> .

Logical Functions

Item	Description
<i>Logical Operators</i>	Logical operators (and, or, not) enable you to logically combine multiple expressions to evaluate a larger, more complex expression whose output is true or false. See <i>Logical Operators</i> .

AND Function	Returns <code>true</code> if both arguments evaluate to <code>true</code> . Equivalent to the <code>&&</code> operator. See <i>AND Function</i> .
OR Function	Returns <code>true</code> if either argument evaluates to <code>true</code> . Equivalent to the <code> </code> operator. See <i>OR Function</i> .
NOT Function	Returns <code>true</code> if the argument evaluates to <code>false</code> , and vice-versa. Equivalent to the <code>!</code> operator. See <i>NOT Function</i> .

Comparison Functions

Item	Description
<i>Comparison Operators</i>	Comparison operators enable you to compare values in the left-hand side of an expression to the values in the right-hand side of an expression. See <i>Comparison Operators</i> .
<i>ISEVEN Function</i>	Returns <code>true</code> if the argument is an even value. Argument can be an Integer, a function returning Integers, or a column reference. See <i>ISEVEN Function</i> .
<i>ISODD Function</i>	Returns <code>true</code> if the argument is an odd value. Argument can be an Integer, a function returning Integers, or a column reference. See <i>ISODD Function</i> .
<i>IN Function</i>	Returns <code>true</code> if the first parameter is contained in the array of values in the second parameter. See <i>IN Function</i> .
<i>MATCHES Function</i>	Returns <code>true</code> if a value contains a string or pattern. The value to search can be a string literal, a function returning a string, or a reference to a column of String type. See <i>MATCHES Function</i> .
<i>EQUAL Function</i>	Returns <code>true</code> if the first argument is equal to the second argument. Equivalent to the <code>=</code> operator. See <i>EQUAL Function</i> .
<i>NOTEQUAL Function</i>	Returns <code>true</code> if the first argument is not equal to the second argument. Equivalent to the <code><></code> or <code>!=</code> operator. See <i>NOTEQUAL Function</i> .
<i>GREATERTHAN Function</i>	Returns <code>true</code> if the first argument is greater than but not equal to the second argument. Equivalent to the <code>></code> operator. See <i>GREATERTHAN Function</i> .
<i>GREATERTHANEQUAL Function</i>	Returns <code>true</code> if the first argument is greater than or equal to the second argument. Equivalent to the <code>>=</code> operator. See <i>GREATERTHANEQUAL Function</i> .
<i>LESSTHAN Function</i>	Returns <code>true</code> if the first argument is less than but not equal to the second argument. Equivalent to the <code><</code> operator. See <i>LESSTHAN Function</i> .
<i>LESSTHANEQUAL Function</i>	Returns <code>true</code> if the first argument is less than or equal to the second argument. Equivalent to the <code><=</code> operator. See <i>LESSTHANEQUAL Function</i> .

Math Functions

Item	Description
<i>Numeric Operators</i>	Numeric operators enable you to generate new values based on a computation (e.g. <code>3 + 4</code>). See <i>Numeric Operators</i> .
<i>NUMFORMAT Function</i>	Formats a numeric set of values according to the specified number formatting. Source values can be a literal value, a function returning a numeric value, or reference to a column containing Integer or Decimal values. See <i>NUMFORMAT Function</i> .
<i>ADD Function</i>	Returns the value of summing the first argument and the second argument. Equivalent to the <code>+</code> operator. See <i>ADD Function</i> .
<i>SUBTRACT Function</i>	Returns the value of subtracting the second argument from the first argument. Equivalent to the <code>-</code> operator. See <i>SUBTRACT Function</i> .
<i>MULTIPLY Function</i>	Returns the value of multiplying the first argument by the second argument. Equivalent to the <code>*</code> operator. See <i>MULTIPLY Function</i> .
<i>DIVIDE Function</i>	Returns the value of dividing the first argument by the second argument. Equivalent to the <code>/</code> operator. See <i>DIVIDE Function</i> .
<i>MOD Function</i>	Returns the modulo value, which is the remainder of dividing the first argument by the second argument. Equivalent to the <code>%</code> operator. See <i>MOD Function</i> .

<i>NEGATE Function</i>	Returns the opposite of the value that is the first argument. Equivalent to the – operator placed in front of the argument. See <i>NEGATE Function</i> .
<i>SIGN Function</i>	Computes the positive or negative sign of a given numeric value. The value can be a Decimal or Integer literal, a function returning Decimal or Integer, or a reference to a column containing numeric values. See <i>SIGN Function</i> .
<i>LCM Function</i>	Returns the least common multiple shared by the first and second arguments. See <i>LCM Function</i> .
<i>ABS Function</i>	Computes the absolute value of a given numeric value. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ABS Function</i> .
<i>EXP Function</i>	Computes the value of <i>e</i> raised to the specified power. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>EXP Function</i> .
<i>LOG Function</i>	Computes the logarithm of the first argument with a base of the second argument. See <i>LOG Function</i> .
<i>LN Function</i>	Computes the natural logarithm of an input value. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>LN Function</i> .
<i>POW Function</i>	Computes the value of the first argument raised to the value of the second argument. See <i>POW Function</i> .
<i>SQRT Function</i>	Computes the square root of the input parameter. Input value can be a Decimal or Integer literal or a reference to a column containing numeric values. All generated values are non-negative. See <i>SQRT Function</i> .
<i>CEILING Function</i>	Computes the ceiling of a value, which is the smallest integer that is greater than the input value. Input can be an Integer, a Decimal, a column reference, or an expression. See <i>CEILING Function</i> .
<i>FLOOR Function</i>	Computes the largest integer that is not more than the input value. Input can be an Integer, a Decimal, a column reference, or an expression. See <i>FLOOR Function</i> .
<i>ROUND Function</i>	Rounds input value to the nearest integer. Input can be an Integer, a Decimal, a column reference, or an expression. Optional second argument can be used to specify the number of digits to which to round. See <i>ROUND Function</i> .
<i>TRUNC Function</i>	Removes all digits to the right of the decimal point for any value. Optionally, you can specify the number of digits to which to round. Input can be an Integer, a Decimal, a column reference, or an expression. See <i>TRUNC Function</i> .

Trigonometry Functions

Item	Description
<i>SIN Function</i>	Computes the sine of an input value for an angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>SIN Function</i> .
<i>COS Function</i>	Computes the cosine of an input value for an angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>COS Function</i> .
<i>TAN Function</i>	Computes the tangent of an input value for an angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>TAN Function</i> .
<i>ASIN Function</i>	For input values between -1 and 1 inclusive, this function returns the angle in radians whose sine value is the input. This function is the inverse of the sine function. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ASIN Function</i> .
<i>ACOS Function</i>	For input values between -1 and 1 inclusive, this function returns the angle in radians whose cosine value is the input. This function is the inverse of the cosine function. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ACOS Function</i> .
<i>ATAN Function</i>	For input values between -1 and 1 inclusive, this function returns the angle in radians whose tangent value is the input. This function is the inverse of the tangent function. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ATAN Function</i> .
<i>SINH Function</i>	Computes the hyperbolic sine of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>SINH Function</i> .
<i>COSH Function</i>	Computes the hyperbolic cosine of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>COSH Function</i> .
<i>TANH Function</i>	Computes the hyperbolic tangent of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>TANH Function</i> .

<i>ASINH Function</i>	Computes the arcsine of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ASINH Function</i> .
<i>ACOSH Function</i>	Computes the arccosine of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ACOSH Function</i> .
<i>ATANH Function</i>	Computes the arctangent of an input value for a hyperbolic angle measured in radians. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>ATANH Function</i> .
<i>DEGREES Function</i>	Computes the degrees of an input value measuring the radians of an angle. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>DEGREES Function</i> .
<i>RADIANS Function</i>	Computes the radians of an input value measuring degrees of an angle. The value can be a Decimal or Integer literal or a reference to a column containing numeric values. See <i>RADIANS Function</i> .

Date Functions

Item	Description
<i>DATE Function</i>	Generates a date value from three inputs of Integer type: year, month, and day. See <i>DATE Function</i> .
<i>TIME Function</i>	Generates time values from three inputs of Integer type: hour, minute, and second. See <i>TIME Function</i> .
<i>DATETIME Function</i>	Generates a Datetime value from the following inputs of Integer type: year, month, day, hour, minute, and second. See <i>DATETIME Function</i> .
<i>DATEADD Function</i>	Add a specified number of units to a valid date. Units can be any supported Datetime unit (e.g. minute, month, year, etc.). Input must be a column reference containing dates. See <i>DATEADD Function</i> .
<i>DATEDIF Function</i>	Calculates the difference between two valid date values for the specified units of measure. See <i>DATEDIF Function</i> .
<i>DATEFORMAT Function</i>	Formats a specified Datetime set of values according to the specified date format. Source values can be a reference to a column containing Datetime values. See <i>DATEFORMAT Function</i> .
<i>UNIXTIMEFORMAT Function</i>	Formats a set of Unix timestamps according to a specified date formatting string. See <i>UNIXTIMEFORMAT Function</i> .
<i>MONTH Function</i>	Derives the month integer value from a Datetime value. Source value can be a reference to a column containing Datetime values or a literal. See <i>MONTH Function</i> .
<i>MONTHNAME Function</i>	Derives the full name from a Datetime value of the corresponding month as a String. Source value can be a reference to a column containing Datetime values or a literal. See <i>MONTHNAME Function</i> .
<i>EOMONTH Function</i>	Returns the serial date number for the last day of the month before or after a specified number of months from a starting date. See <i>EOMONTH Function</i> .
<i>YEAR Function</i>	Derives the four-digit year value from a Datetime value. Source value can be a reference to a column containing Datetime values or a literal. See <i>YEAR Function</i> .
<i>DAY Function</i>	Derives the numeric day value from a Datetime value. Source value can be a reference to a column containing Datetime values or a literal. See <i>DAY Function</i> .
<i>WEEKNUM Function</i>	Derives the numeric value for the week within the year (1, 2, etc.). Input must be the output of the <i>DATE</i> function or a reference to a column containing Datetime values. The output of this function increments on Sunday. See <i>WEEKNUM Function</i> .
<i>WEEKDAY Function</i>	Derives the numeric value for the day of the week (1, 2, etc.). Input must be a reference to a column containing Datetime values. See <i>WEEKDAY Function</i> .
<i>WEEKDAYNAME Function</i>	Derives the full name from a Datetime value of the corresponding weekday as a String. Source value can be a reference to a column containing Datetime values or a literal. See <i>WEEKDAYNAME Function</i> .
<i>HOURLY Function</i>	Derives the hour value from a Datetime value. Generated hours are expressed according to the 24-hour clock. See <i>HOURLY Function</i> .
<i>MINUTE Function</i>	Derives the minutes value from a Datetime value. Minutes are expressed as integers from 0 to 59. See <i>MINUTE Function</i> .
<i>SECOND Function</i>	Derives the seconds value from a Datetime value. Source value can be a reference to a column containing Datetime values or a literal. See <i>SECOND Function</i> .

<i>UNIXTIME Function</i>	Derives the Unixtime (or epoch time) value from a Datetime value. Source value can be a reference to a column containing Datetime values. See <i>UNIXTIME Function</i> .
<i>NOW Function</i>	Derives the timestamp for the current time in UTC time zone. You can specify a different time zone by optional parameter. See <i>NOW Function</i> .
<i>TODAY Function</i>	Derives the value for the current date in UTC time zone. You can specify a different time zone by optional parameter. See <i>TODAY Function</i> .
<i>PARSEDATE Function</i>	Evaluates an input against the default input formats or (if specified) an array of Datetime format strings in their listed order. If the input matches one of the formats, the function outputs a Datetime value. See <i>PARSEDATE Function</i> .
<i>NETWORKDAY S Function</i>	Calculates the number of working days between two specified dates, assuming Monday - Friday workweek. Optional list of holidays can be specified. See <i>NETWORKDAYS Function</i> .
<i>NETWORKDAY SINTL Function</i>	Calculates the number of working days between two specified dates. Optionally, you can specify which days of the week are working days as an input parameter. Optional list of holidays can be specified. See <i>NETWORKDAYSINTL Function</i> .
<i>MINDATE Function</i>	Computes the minimum value found in all row values in a Datetime column. See <i>MINDATE Function</i> .
<i>MAXDATE Function</i>	Computes the maximum value found in all row values in a Datetime column. See <i>MAXDATE Function</i> .
<i>MODEDATE Function</i>	Computes the most frequent (mode) value found in all row values in a Datetime column. See <i>MODEDATE Function</i> .
<i>WORKDAY Function</i>	Calculates the work date that is before or after a start date, as specified by a number of days. A set of holiday dates can be optionally specified. See <i>WORKDAY Function</i> .
<i>WORKDAYINTL Function</i>	Calculates the work date that is before or after a start date, as specified by a number of days. You can also specify which days of the week are working days and a list of holidays via parameters. See <i>WORKDAYINTL Function</i> .
<i>CONVERTFRO MUTC Function</i>	Converts Datetime value to corresponding value of the specified time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values. See <i>CONVERTFROMUTC Function</i> .
<i>CONVERTTOU TC Function</i>	Converts Datetime value in specified time zone to corresponding value in UTC time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values. See <i>CONVERTTOUTC Function</i> .
<i>CONVERTTIME ZONE Function</i>	Converts Datetime value in specified time zone to corresponding value second specified time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values. See <i>CONVERTTIMEZONE Function</i> .
<i>MINDATEIF Function</i>	Returns the minimum Datetime value of rows in each group that meet a specific condition. Set of values must valid Datetime values. See <i>MINDATEIF Function</i> .
<i>MAXDATEIF Function</i>	Returns the maximum Datetime value of rows in each group that meet a specific condition. Set of values must valid Datetime values. See <i>MAXDATEIF Function</i> .
<i>MODEDATEIF Function</i>	Returns the most common Datetime value of rows in each group that meet a specific condition. Set of values must valid Datetime values. See <i>MODEDATEIF Function</i> .
<i>KTHLARGESTD ATE Function</i>	Extracts the ranked Datetime value from the values in a column, where $k=1$ returns the maximum value. The value for k must be between 1 and 1000, inclusive. Inputs must be valid Datetime values. See <i>KTHLARGESTDATE Function</i> .
<i>KTHLARGESTU NIQUEDATE Function</i>	Extracts the ranked unique Datetime value from the values in a column, where $k=1$ returns the maximum value. The value for k must be between 1 and 1000, inclusive. Inputs must be Datetime. See <i>KTHLARGESTUNIQUEDATE Function</i> .
<i>KTHLARGESTU NIQUEDATEIF Function</i>	Extracts the ranked unique Datetime value from the values in a column, where $k=1$ returns the maximum value, when a specified condition is met. The value for k must be between 1 and 1000, inclusive. Inputs must be Datetime. See <i>KTHLARGESTUNIQUEDATEIF Function</i> .
<i>KTHLARGESTD ATEIF Function</i>	Extracts the ranked Datetime value from the values in a column, where $k=1$ returns the maximum value, when a specified condition is met. The value for k must be between 1 and 1000, inclusive. Inputs must be Datetime. See <i>KTHLARGESTDATEIF Function</i> .
<i>SERIALNUMBE R Function</i>	Generates a serial date number from a valid date value. See <i>SERIALNUMBER Function</i> .

String Functions

Item	Description
<i>CHAR Function</i>	Generates the Unicode character corresponding to an inputted Integer value. See <i>CHAR Function</i> .
<i>UNICODE Function</i>	Generates the Unicode index value for the first character of the input string. See <i>UNICODE Function</i> .
<i>UPPER Function</i>	All alphabetical characters in the input value are converted to uppercase in the output value. See <i>UPPER Function</i> .
<i>LOWER Function</i>	All alphabetical characters in the input value are converted to lowercase in the output value. See <i>LOWER Function</i> .
<i>PROPER Function</i>	Converts an input string to propercase. Input can be a column reference or a string literal. See <i>PROPER Function</i> .
<i>TRIM Function</i>	Removes leading and trailing whitespace from a string. Spacing between words is not removed. See <i>TRIM Function</i> .
<i>REMOVEWHITESPACE Function</i>	Removes all whitespace from a string, including leading and trailing whitespace and all whitespace within the string. See <i>REMOVEWHITESPACE Function</i> .
<i>REMOVESYMBOLS Function</i>	Removes all characters from a string that are not letters, numbers, accented Latin characters, or whitespace. See <i>REMOVESYMBOLS Function</i> .
<i>LEN Function</i>	Returns the number of characters in a specified string. String value can be a column reference or string literal. See <i>LEN Function</i> .
<i>FIND Function</i>	Returns the index value in the input string where a specified matching string is located in provided column, string literal, or function returning a string. Search is conducted left-to-right. See <i>FIND Function</i> .
<i>RIGHTFIND Function</i>	Returns the index value in the input string where the last instance of a matching string is located. Search is conducted right-to-left. See <i>RIGHTFIND Function</i> .
<i>SUBSTRING Function</i>	Matches some or all of a string, based on the user-defined starting and ending index values within the string. See <i>SUBSTRING Function</i> .
<i>SUBSTITUTE Function</i>	Replaces found string literal or pattern or column with a string, column, or function returning strings. See <i>SUBSTITUTE Function</i> .
<i>LEFT Function</i>	Matches the leftmost set of characters in a string, as specified by parameter. The string can be specified as a column reference or a string literal. See <i>LEFT Function</i> .
<i>RIGHT Function</i>	Matches the right set of characters in a string, as specified by parameter. The string can be specified as a column reference or a string literal. See <i>RIGHT Function</i> .
<i>PAD Function</i>	Pads string values to be a specified minimum length by adding a designated character to the left or right end of the string. Returned value is of String type. See <i>PAD Function</i> .
<i>MERGE Function</i>	Merges two or more columns of String type to generate output of String type. Optionally, you can insert a delimiter between the merged values. See <i>MERGE Function</i> .
<i>STARTSWITH Function</i>	Returns <code>true</code> if the leftmost set of characters of a column of values matches a pattern. The source value can be any data type, and the pattern can be a Pattern , regular expression, or a string. See <i>STARTSWITH Function</i> .
<i>ENDSWITH Function</i>	Returns <code>true</code> if the rightmost set of characters of a column of values matches a pattern. The source value can be any data type, and the pattern can be a Pattern , regular expression, or a string. See <i>ENDSWITH Function</i> .
<i>REPEAT Function</i>	Repeats a string a specified number of times. The string can be specified as a String literal, a function returning a String, or a column reference. See <i>REPEAT Function</i> .
<i>EXACT Function</i>	Returns <code>true</code> if the second string evaluates to be an exact match of the first string. Source values can be string literals, column references, or expressions that evaluate to strings. See <i>EXACT Function</i> .
<i>STRINGGREATERTHAN Function</i>	Returns <code>true</code> if the first string evaluates to be greater than the second string, based on a set of common collation rules. See <i>STRINGGREATERTHAN Function</i> .
<i>STRINGGREATERTHANEQUAL Function</i>	Returns <code>true</code> if the first string evaluates to be greater than or equal to the second string, based on a set of common collation rules. See <i>STRINGGREATERTHANEQUAL Function</i> .
<i>STRINGLESSTHAN Function</i>	Returns <code>true</code> if the first string evaluates to be less than the second string, based on a set of common collation rules. See <i>STRINGLESSTHAN Function</i> .
<i>STRINGLESSTHANEQUAL Function</i>	Returns <code>true</code> if the first string evaluates to be less than or equal to the second string, based on a set of common collation rules. See <i>STRINGLESSTHANEQUAL Function</i> .

<i>DOUBLEMETAPHONE Function</i>	Returns a two-element array of primary and secondary phonetic encodings for an input string, based on the Double Metaphone algorithm. See <i>DOUBLEMETAPHONE Function</i> .
<i>DOUBLEMETAPHONEEQUALS Function</i>	Compares two input strings using the Double Metaphone algorithm. An optional threshold parameter can be modified to adjust the tolerance for matching. See <i>DOUBLEMETAPHONEEQUALS Function</i> .
<i>TRANSLITERATE Function</i>	Transliterates Asian script characters from one script form to another. The string can be specified as a column reference or a string literal. See <i>TRANSLITERATE Function</i> .
<i>TRIMQUOTES Function</i>	Removes leading and trailing quotes or double-quotes from a string. Quote marks in the middle of the string are not removed. See <i>TRIMQUOTES Function</i> .
<i>BASE64ENCODE Function</i>	Converts an input value to base64 encoding with optional padding with an equals sign (=). Input can be of any type. Output type is String. See <i>BASE64ENCODE Function</i> .
<i>BASE64DECODE Function</i>	Converts an input base64 value to text. Output type is String. See <i>BASE64DECODE Function</i> .

Nested Functions

Item	Description
<i>ARRAYCONCAT Function</i>	Combines the elements of one array with another, listing all elements of the first array before listing all elements of the second array. See <i>ARRAYCONCAT Function</i> .
<i>ARRAYCROSS Function</i>	Generates a nested array containing the cross-product of all elements in two or more arrays. See <i>ARRAYCROSS Function</i> .
<i>ARRAYELEMENTAT Function</i>	Computes the 0-based index value for an array element in the specified column, array literal, or function that returns an array. See <i>ARRAYELEMENTAT Function</i> .
<i>ARRAYINDEXOF Function</i>	Computes the index at which a specified element is first found within an array. Indexing is left to right. See <i>ARRAYINDEXOF Function</i> .
<i>ARRAYINTERSECT Function</i>	Generates an array containing all elements that appear in multiple input arrays, referenced as column names or array literals. See <i>ARRAYINTERSECT Function</i> .
<i>ARRAYLEN Function</i>	Computes the number of elements in the arrays in the specified column, array literal, or function that returns an array. See <i>ARRAYLEN Function</i> .
<i>ARRAYMERGEELEMENTS Function</i>	Merges the elements of an array in left to right order into a string. Values are optionally delimited by a provided delimiter. See <i>ARRAYMERGEELEMENTS Function</i> .
<i>ARRAYRIGHTINDEXOF Function</i>	Computes the index at which a specified element is first found within an array, when searching right to left. Returned value is based on left-to-right indexing. See <i>ARRAYRIGHTINDEXOF Function</i> .
<i>ARRAYSLICE Function</i>	Returns an array containing a slice of the input array, as determined by starting and ending index parameters. See <i>ARRAYSLICE Function</i> .
<i>ARRAYSORT Function</i>	Sorts array values in the specified column, array literal, or function that returns an array in ascending or descending order. See <i>ARRAYSORT Function</i> .
<i>ARRAYSTOMAP Function</i>	Combines one array containing keys and another array containing values into an Object of key-value pairs. See <i>ARRAYSTOMAP Function</i> .
<i>ARRAYUNIQUE Function</i>	Generates an array of all unique elements among one or more arrays. See <i>ARRAYUNIQUE Function</i> .
<i>ARRAYZIP Function</i>	Combines multiple arrays into a single nested array, with element 1 of array 1 paired with element 2 of array 2 and so on. Arrays are expressed as column names or as array literals. See <i>ARRAYZIP Function</i> .
<i>FILTEROBJECT Function</i>	Filters the keys and values from an Object data type column based on a specified key value. See <i>FILTEROBJECT Function</i> .
<i>KEYS Function</i>	Extracts the key values from an Object data type column and stores them in an array of String values. See <i>KEYS Function</i> .
<i>LISTAVERAGE Function</i>	Computes the average of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTAVERAGE Function</i> .
<i>LISTMAX Function</i>	Computes the maximum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTMAX Function</i> .

<i>LISTMIN Function</i>	Computes the minimum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTMIN Function</i> .
<i>LISTMODE Function</i>	Computes the most common value of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTMODE Function</i> .
<i>LISTSTDEV Function</i>	Computes the standard deviation of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTSTDEV Function</i> .
<i>LISTSUM Function</i>	Computes the sum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTSUM Function</i> .
<i>LISTVAR Function</i>	Computes the variance of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type. See <i>LISTVAR Function</i> .

Type Functions

Item	Description
<i>NULL Function</i>	The <i>NULL</i> function generates null values. See <i>NULL Function</i> .
<i>IFNULL Function</i>	The <i>IFNULL</i> function writes out a specified value if the source value is a null. Otherwise, it writes the source value. Input can be a literal, a column reference, or a function. See <i>IFNULL Function</i> .
<i>IFMISSING Function</i>	The <i>IFMISSING</i> function writes out a specified value if the source value is a null or missing value. Otherwise, it writes the source value. Input can be a literal, a column reference, or a function. See <i>IFMISSING Function</i> .
<i>IFMISMATCHED Function</i>	The <i>IFMISMATCHED</i> function writes out a specified value if the input expression does not match the specified data type or typing array. Otherwise, it writes the source value. Input can be a literal, a column reference, or a function. See <i>IFMISMATCHED Function</i> .
<i>IFVALID Function</i>	The <i>IFVALID</i> function writes out a specified value if the input expression matches the specified data type. Otherwise, it writes the source value. Input can be a literal, a column reference, or a function. See <i>IFVALID Function</i> .
<i>ISNULL Function</i>	The <i>ISNULL</i> function tests whether a column of values contains null values. For input column references, this function returns <i>true</i> or <i>false</i> . See <i>ISNULL Function</i> .
<i>ISMISSING Function</i>	The <i>ISMISSING</i> function tests whether a column of values is missing or null. For input column references, this function returns <i>true</i> or <i>false</i> . See <i>ISMISSING Function</i> .
<i>ISMISMATCHED Function</i>	Tests whether a set of values is not valid for a specified data type. See <i>ISMISMATCHED Function</i> .
<i>VALID Function</i>	Tests whether a set of values is valid for a specified data type and is not a null value. See <i>VALID Function</i> .
<i>PARSEINT Function</i>	Evaluates a String input against the Integer datatype. If the input matches, the function outputs an Integer value. Input can be a literal, a column of values, or a function returning String values. See <i>PARSEINT Function</i> .
<i>PARSEBOOLEAN Function</i>	Evaluates a String input against the Boolean datatype. If the input matches, the function outputs a Boolean value. Input can be a literal, a column of values, or a function returning String values. See <i>PARSEBOOLEAN Function</i> .
<i>PARSEFLOAT Function</i>	Evaluates a String input against the Decimal datatype. If the input matches, the function outputs a Decimal value. Input can be a literal, a column of values, or a function returning String values. See <i>PARSEFLOAT Function</i> .

Window Functions

Item	Description
<i>PREV Function</i>	Extracts the value from a column that is a specified number of rows before the current value. See <i>PREV Function</i> .
<i>NEXT Function</i>	Extracts the value from a column that is a specified number of rows after the current value. See <i>NEXT Function</i> .
<i>FILL Function</i>	Fills any missing or null values in the specified column with the most recent non-blank value, as determined by the specified window of rows before and after the blank value. See <i>FILL Function</i> .

<i>RANK Function</i>	Computes the rank of an ordered set of value within groups. Tie values are assigned the same rank, and the next ranking is incremented by the number of tie values. See <i>RANK Function</i> .
<i>DENSERANK Function</i>	Computes the rank of an ordered set of value within groups. Tie values are assigned the same rank, and the next ranking is incremented by 1. See <i>DENSERANK Function</i> .
<i>ROLLINGAVERAGE Function</i>	Computes the rolling average of values forward or backward of the current row within the specified column. See <i>ROLLINGAVERAGE Function</i> .
<i>ROLLINGMODE Function</i>	Computes the rolling mode (most common value) forward or backward of the current row within the specified column. Input values can be Integer, Decimal, or Datetime data type. See <i>ROLLINGMODE Function</i> .
<i>ROLLINGMAX Function</i>	Computes the rolling maximum of values forward or backward of the current row within the specified column. Inputs can be Integer, Decimal, or Datetime. See <i>ROLLINGMAX Function</i> .
<i>ROLLINGMIN Function</i>	Computes the rolling minimum of values forward or backward of the current row within the specified column. Inputs can be Integer, Decimal, or Datetime. See <i>ROLLINGMIN Function</i> .
<i>ROLLINGMODEDATE Function</i>	Computes the rolling mode (most common value) forward or backward of the current row within the specified column. Input values must be of Datetime data type. See <i>ROLLINGMODEDATE Function</i> .
<i>ROLLINGMAXDATE Function</i>	Computes the rolling maximum of date values forward or backward of the current row within the specified column. Inputs must be of Datetime type. See <i>ROLLINGMAXDATE Function</i> .
<i>ROLLINGMINDATE Function</i>	Computes the rolling minimum of Date values forward or backward of the current row within the specified column. Inputs must be of Datetime type. See <i>ROLLINGMINDATE Function</i> .
<i>ROLLINGSUM Function</i>	Computes the rolling sum of values forward or backward of the current row within the specified column. See <i>ROLLINGSUM Function</i> .
<i>ROLLINGSTDEV Function</i>	Computes the rolling standard deviation of values forward or backward of the current row within the specified column. See <i>ROLLINGSTDEV Function</i> .
<i>ROLLINGSTDEVSAMP Function</i>	Computes the rolling standard deviation of values forward or backward of the current row within the specified column using the sample statistical method. See <i>ROLLINGSTDEVSAMP Function</i> .
<i>ROLLINGVAR Function</i>	Computes the rolling variance of values forward or backward of the current row within the specified column. See <i>ROLLINGVAR Function</i> .
<i>ROLLINGVARSAMP Function</i>	Computes the rolling variance of values forward or backward of the current row within the specified column using the sample statistical method. See <i>ROLLINGVARSAMP Function</i> .
<i>ROLLINGCOUNTA Function</i>	Computes the rolling count of non-null values forward or backward of the current row within the specified column. See <i>ROLLINGCOUNTA Function</i> .
<i>ROLLINGKTHLARGEST Function</i>	Computes the rolling <i>k</i> th largest value forward or backward of the current row. Inputs can be Integer, Decimal, or Datetime. See <i>ROLLINGKTHLARGEST Function</i> .
<i>ROLLINGKTHLARGESTUNIQUE Function</i>	Computes the rolling unique <i>k</i> th largest value forward or backward of the current row. Inputs can be Integer, Decimal, or Datetime. See <i>ROLLINGKTHLARGESTUNIQUE Function</i> .
<i>ROLLINGLIST Function</i>	Computes the rolling list of values forward or backward of the current row within the specified column and returns an array of these values. See <i>ROLLINGLIST Function</i> .
<i>ROWNUMBER Function</i>	Generates a new column containing the row number as sorted by the <code>order</code> parameter and optionally grouped by the <code>group</code> parameter. See <i>ROWNUMBER Function</i> .
<i>SESSION Function</i>	Generates a new session identifier based on a sorted column of timestamps and a specified rolling timeframe. See <i>SESSION Function</i> .

Other Functions

Item	Description
<i>COALESCE Function</i>	Function returns the first non-missing value found in an array of columns. See <i>COALESCE Function</i> .
<i>RAND Function</i>	The <code>RAND</code> function generates a random real number between 0 and 1. The function accepts an optional integer parameter, which causes the same set of random numbers to be generated with each job execution. See <i>RAND Function</i> .
<i>RANDBETWEEN Function</i>	Generates a random integer between a low and a high number. Two inputs may be Integer or Decimal types, functions returning these types, or column references. See <i>RANDBETWEEN Function</i> .
<i>PI Function</i>	The <code>PI</code> function generates the value of pi to 15 decimal places: 3.1415926535897932. See <i>PI Function</i> .

<i>SOURCEROW NUMBER Function</i>	Returns the row number of the current row as it appeared in the original source dataset before any steps had been applied. See <i>SOURCEROWNUMBER Function</i> .
<i>IF Function</i>	The IF function allows you to build if/then/else conditional logic within your transforms. See <i>IF Function</i> .
<i>CASE Function</i>	The CASE function allows you to perform multiple conditional tests on a set of expressions within a single statement. When a test evaluates to <code>true</code> , a corresponding output is generated. Outputs may be a literal or expression. See <i>CASE Function</i> .
<i>Ternary Operators</i>	Ternary operators allow you to build if/then/else conditional logic within your transforms. Please use the IF function instead. See <i>Ternary Operators</i> .
<i>IPTOINT Function</i>	Computes an integer value for a four-octet internet protocol (IP) address. Source value must be a valid IP address or a column reference to IP addresses. See <i>IPTOINT Function</i> .
<i>IPFROMINT Function</i>	Computes a four-octet internet protocol (IP) address from a 32-bit integer input. See <i>IPFROMINT Function</i> .
<i>RANGE Function</i>	Computes an array of integers, from a beginning integer to an end (stop) integer, stepping by a third parameter. See <i>RANGE Function</i> .
<i>HOST Function</i>	Finds the host value from a valid URL. Input values must be of URL or String type and can be literals or column references. See <i>HOST Function</i> .
<i>DOMAIN Function</i>	Finds the value for the domain from a valid URL. Input values must be of URL or String type. See <i>DOMAIN Function</i> .
<i>SUBDOMAIN Function</i>	Finds the value a subdomain value from a valid URL. Input values must be of URL or String type. See <i>SUBDOMAIN Function</i> .
<i>SUFFIX Function</i>	Finds the suffix value after the domain from a valid URL. Input values must be of URL or String type. See <i>SUFFIX Function</i> .
<i>URLPARAMS Function</i>	Extracts the query parameters of a URL into an Object. The Object keys are the parameter's names, and its values are the parameter's values. Input values must be of URL or String type. See <i>URLPARAMS Function</i> .