

Table of Contents

- Expressions** 1
 - Constants 1
 - Operators 1
 - Logical Functions 2
 - Math Functions 2
 - String Functions 3
 - Date-time Functions 3

Expressions

An expression is a string that, when parsed and processed, evaluates some value. Expressions consist of column/field names, constants, operators, and functions. Column/field names must be wrapped with brackets. To learn more about available operators and functions, see below. The following are examples of regular expressions.

```
"[Quantity] * [UnitPrice] * (1 - [BonusAmount])"
```

```
"[FirstName] + ' ' + [LastName]"
```

Boolean expressions:

```
"[Country] == 'USA' "
```

```
"[OrderDate] > #8/16/1994# AND [Quantity] > 20"
```

Constants

| Constant | Description | Example |
|---------------------|---|---|
| String constants | Wrap string constants in apostrophes. If a string contains an apostrophe, double the apostrophe. | [Country] == 'France' [Name] == 'ONeil' |
| Date-time constants | Wrap date-time constants in '#'. #MM/DD/YYYY# | [OrderDate] >= #2018-03-22 13:18:51.94944# |
| True | Represents the Boolean True value. | [InStock] == True |
| False | Represents the Boolean False value. | [InStock] == False |
| Numeric | Specify different numeric constant types in a string form using suffixes Int32 (int) - 1 Int16 (short) - 1s Byte (byte) - 1b Double (double) - 1.0 Single (float) - 1.0f Decimal (decimal) - 1.0m | [Price] == 25.0m |

Operators

| Operator | Description | Example |
|----------|---|--|
| + | Adds the value of one numeric expression to another or concatenates two strings. | [UnitPrice] + 4 [FirstName] + ' ' + [LastName] |
| - | Finds the difference between two numbers. | [Price1] - [Price2] |
| * | Multiplies the value of two expressions. | [Quantity] * [UnitPrice] * (1 - [BonusAmount]) |
| / | Divides the first operand by the second. | [Quantity] / 2 |
| % | Returns the remainder (modulus) obtained by dividing one numeric expression by another. | [Quantity] % 3 |

| Operator | Description | Example |
|-------------|--|-------------------------------------|
| == = | Returns true if both operands have the same value; otherwise, it returns false. | [Quantity] == 10 |
| != | Returns true if the operands do not have the same value; otherwise, it returns false. | [Country] != 'France' |
| < | Less than operator. Used to compare expressions. | [UnitPrice] < 20 |
| ≤ | Less than or equal to operator. Used to compare expressions. | [UnitPrice] ≤ 20 |
| >= | Greater than or equal to operator. Used to compare expressions. | [UnitPrice] >= 30 |
| > | Greater than operator. Used to compare expressions. | [UnitPrice] > 30 |
| In (,,) | Tests for the existence of a property in an object. | [Country] In ('USA', 'UK', 'Italy') |
| Between (,) | Specifies a range to test. Returns true if a value is greater than or equal to the first operand and less than or equal to the second operand. | [Quantity] Between (10, 20) |
| Not ! | Performs a logical negation on a Boolean expression. | Not [InStock] ![InStock] |
| Is Null | Returns true if an expression is a null reference, the one that does not refer to any object. | [Region] is null |

Logical Functions

| Function | Description | Example |
|---|--|--|
| lif(Expression1, True_Value1, ..., ExpressionN, True_ValueN, False_Value) | Returns one of several specified values depending upon the values of logical expressions. The function can take 2N+1 arguments (N - the number of specified logical expressions): Each odd argument specifies a logical expression; Each even argument specifies the value that is returned if the previous expression evaluates to true; | lif(Name = 'Bob', 1, 0) lif(Name = 'Bob', 1, Name = 'Dan', 2, Name = 'Sam', 3, 0) |
| IsNull(Value) | Returns True if the specified Value is NULL. | IsNull([OrderDate]) |
| IsNull(Value1, Value2) | Returns Value1 if it is not set to NULL; otherwise, Value2 is returned. | IsNull([ShipDate], [RequiredDate]) |
| IsNullOrEmpty(String) | Returns True if the specified String object is NULL or an empty string; otherwise, False is returned. | IsNullOrEmpty([ProductName]) |

Math Functions

| Function | Description | Example |
|---------------------|--|-------------------------|
| Max(Value1, Value2) | Returns the maximum value from the specified values. | Max([Value1], [Value2]) |

| Function | Description | Example |
|-------------------------|--|-------------------------|
| Min(Value1, Value2) | Returns the minimum value from the specified values. | Min([Value1], [Value2]) |
| Round(Value) | Rounds the given value to the nearest integer. | Round([Value]) |
| Round(Value, Precision) | Rounds the given value to the nearest integer, or to a specified number of decimal places. | Round([Value], 2) |

String Functions

| Function | Description | Example |
|--|--|----------------------------------|
| Concat(String1, ... , StringN) | Returns a string value containing the concatenation of the current string with any additional strings. | Concat('A', '), [ProductName]) |
| Contains(String1, SubString1) | Returns True if SubString1 occurs within String1; otherwise, False is returned. | Contains([ProductName], 'dairy') |
| Insert(String1, StartPosition, String2) | Inserts String2 into String1 at the position specified by StartPositon | Insert([Name], 0, 'ABC-') |
| Lower(String) | Returns String in lowercase. | Lower([ProductName]) |
| Replace(String, SubString2, String3) | Returns a copy of String1, in which SubString2 has been replaced with String3. | Replace([Name], 'The ', '') |
| StartsWith(String1, SubString1) | Returns True if the beginning of String1 matches SubString1 otherwise, False. | StartsWith([Title], 'The best') |
| Substring(String, StartPosition, Length) | Retrieves a substring from String. The substring starts at StartPosition and has a specified Length. | Substring([Description], 2, 3) |
| Substring(String, StartPosition) | Retrieves a substring from String. The substring starts at StartPosition. | Substring([Description], 2) |
| Upper(String) | Returns String in uppercase. | Upper([ProductName]) |

Date-time Functions

| Function | Description | Example |
|--------------------------------|--|--------------------------|
| AddDays(DateTime, DaysCount) | Returns a date-time value that is the specified number of days from the specified DateTime. | AddDays([OrderDate], 30) |
| AddHours(DateTime, HoursCount) | Returns a date-time value that is the specified number of hours from the specified DateTime. | AddHours([StartTime], 2) |

| Function | Description | Example |
|--|---|------------------------------------|
| AddMilliseconds(DateTime, MilliSecondsCount) | Returns a date-time value that is the specified number of milliseconds from the specified DateTime. | AddMilliseconds([StartTime], 5000) |
| AddMinutes(DateTime, MinutesCount) | Returns a date-time value that is the specified number of minutes from the specified DateTime. | AddMinutes([StartTime], 30) |
| AddMonths(DateTime, MonthsCount) | Returns a date-time value that is the specified number of months from the specified DateTime. | AddMonths([OrderDate], 1) |
| AddSeconds(DateTime, SecondsCount) | Returns a date-time value that is the specified number of seconds from the specified DateTime. | AddSeconds([StartTime], 60) |
| AddYears(DateTime, YearsCount) | Returns a date-time value that is the specified number of years from the specified DateTime. | AddYears([EndDate], -1) |
| DateDiffDay(startDate, endDate) | Returns the number of day boundaries between two non-nullable dates. | DateDiffDay([StartTime], Now()) |
| DateDiffHour(startDate, endDate) | Returns the number of hour boundaries between two non-nullable dates. | DateDiffHour([StartTime], Now()) |

| Function | Description | Example |
|---|--|---|
| DateDiffMilliSecond(startDate, endDate) | Returns the number of millisecond boundaries between two non-nullable dates. | DateDiffMilliSecond([StartTime], Now()) |
| DateDiffMinute(startDate, endDate) | Returns the number of minute boundaries between two non-nullable dates. | DateDiffMinute([StartTime], Now()) |
| DateDiffMonth(startDate, endDate) | Returns the number of month boundaries between two non-nullable dates. | DateDiffMonth([StartTime], Now()) |
| DateDiffSecond(startDate, endDate) | Returns the number of second boundaries between two non-nullable dates. | DateDiffSecond([StartTime], Now()) |
| DateDiffYear(startDate, endDate) | Returns the number of year boundaries between two non-nullable dates. | DateDiffYear([StartTime], Now()) |
| GetDate(DateTime) | Extracts a date from the defined DateTime. | GetDate([OrderDateTime]) |
| GetDay(DateTime) | Extracts a day from the defined DateTime. | GetDay([OrderDate]) |
| GetDayOfWeek(DateTime) | Extracts a day of the week from the defined DateTime. | GetDayOfWeek([OrderDate]) |
| GetDayOfYear(DateTime) | Extracts a day of the year from the defined DateTime. | GetDayOfYear([OrderDate]) |
| GetHour(DateTime) | Extracts an hour from the defined DateTime. | GetHour([StartTime]) |

| Function | Description | Example |
|--------------------------|---|-----------------------------|
| GetMilliSecond(DateTime) | Extracts milliseconds from the defined DateTime. | GetMilliSecond([StartTime]) |
| GetMinute(DateTime) | Extracts minutes from the defined DateTime. | GetMinute([StartTime]) |
| GetMonth(DateTime) | Extracts a month from the defined DateTime. | GetMonth([StartTime]) |
| GetSecond(DateTime) | Extracts seconds from the defined DateTime. | GetSecond([StartTime]) |
| GetYear(DateTime) | Extracts a year from the defined DateTime. | GetYear([StartTime]) |
| IsApril(DateTime) | Returns True if the specified date falls within April. | IsApril([OrderDate]) |
| IsAugust(DateTime) | Returns True if the specified date falls within August. | IsAugust([OrderDate]) |
| IsDecember(DateTime) | Returns True if the specified date falls within December. | IsDecember([OrderDate]) |
| IsFebruary(DateTime) | Returns True if the specified date falls within February. | IsFebruary([OrderDate]) |
| IsJanuary(DateTime) | Returns True if the specified date falls within January. | IsJanuary([OrderDate]) |
| IsJuly(DateTime) | Returns True if the specified date falls within July. | IsJuly([OrderDate]) |
| IsJune(DateTime) | Returns True if the specified date falls within June. | IsJune([OrderDate]) |
| IsLastMonth(DateTime) | Returns True if the specified date falls within the previous month. | IsLastMonth([OrderDate]) |

| Function | Description | Example |
|-----------------------|--|--------------------------|
| IsLastYear(DateTime) | Returns True if the specified date falls within the previous year. | IsLastYear([OrderDate]) |
| IsMarch(DateTime) | Returns True if the specified date falls within March. | IsMarch([OrderDate]) |
| IsMay(DateTime) | Returns True if the specified date falls within May. | IsMay([OrderDate]) |
| IsNextMonth(DateTime) | Returns True if the specified date falls within the next month. | IsNextMonth([OrderDate]) |
| IsNextYear(DateTime) | Returns True if the specified date falls within the next year. | IsNextYear([OrderDate]) |
| IsNovember(DateTime) | Returns True if the specified date falls within November. | IsNovember([OrderDate]) |
| IsOctober(DateTime) | Returns True if the specified date falls within October. | IsOctober([OrderDate]) |
| IsSameDay(DateTime) | Returns True if the specified date/time values fall within the same day. | IsSameDay([OrderDate]) |
| IsSeptember(DateTime) | Returns True if the specified date falls within September. | IsSeptember([OrderDate]) |
| IsThisMonth(DateTime) | Returns True if the specified date falls within the current month. | IsThisMonth([OrderDate]) |
| IsThisWeek(DateTime) | Returns True if the specified date falls within the current week. | IsThisWeek([OrderDate]) |

| Function | Description | Example |
|---------------------------------|--|---|
| IsYearToDate(DateTime) | Returns True if the specified date falls within the year-to-date period. This period starts from the first day of the current year and continues to the current date (including the current date). | IsYearToDate([OrderDate]) |
| IsThisYear(DateTime) | Returns True if the specified date falls within the current year. | IsThisYear([OrderDate]) |
| LocalDateTimeDayAfterTomorrow() | Returns a date-time value corresponding to the day after Tomorrow. | AddDays(LocalDateTimeDayAfterTomorrow(), 5) |
| LocalDateTimeLastMonth() | Returns the DateTime value corresponding to the first day of the previous month. | AddMonths(LocalDateTimeLastMonth(), 5) |
| LocalDateTimeLastWeek() | Returns a date-time value corresponding to the first day of the previous week. | AddDays(LocalDateTimeLastWeek(), 5) |
| LocalDateTimeLastYear() | Returns the DateTime value corresponding to the first day of the previous year. | AddYears(LocalDateTimeLastYear(), 5) |
| LocalDateTimeNextMonth() | Returns a date-time value corresponding to the first day of the next month. | AddMonths(LocalDateTimeNextMonth(), 5) |
| LocalDateTimeNextWeek() | Returns a date-time value corresponding to the first day of the following week. | AddDays(LocalDateTimeNextWeek(), 5) |

| Function | Description | Example |
|------------------------------|---|---|
| LocalDateTimeNextYear() | Returns a date-time value corresponding to the first day of the following year. | AddYears(LocalDateTimeNextYear(), 5) |
| LocalDateTimeNow() | Returns a date-time value corresponding to the current moment in time. | AddDays(LocalDateTimeNow(), 5) |
| LocalDateTimeThisMonth() | Returns a date-time value corresponding to the first day of the current month. | AddMonths(LocalDateTimeThisMonth(), 5) |
| LocalDateTimeThisWeek() | Returns a date-time value corresponding to the first day of the current week. | AddDays(LocalDateTimeThisWeek(), 5) |
| LocalDateTimeThisYear() | Returns a date-time value corresponding to the first day of the current year. | AddYears(LocalDateTimeThisYear(), 5) |
| LocalDateTimeToday() | Returns a date-time value corresponding to Today. | AddDays(LocalDateTimeToday(), 5) |
| LocalDateTimeTomorrow() | Returns a date-time value corresponding to Tomorrow. | AddDays(LocalDateTimeTomorrow(), 5) |
| LocalDateTimeTwoMonthsAway() | Returns the DateTime value corresponding to the first day of the following month. | AddMonths(LocalDateTimeTwoMonthAway(), 5) |
| LocalDateTimeTwoWeeksAway() | Returns the DateTime value corresponding to the first day of the following week. | AddDays(LocalDateTimeTwoWeeksAway(), 5) |

| Function | Description | Example |
|--------------------------------|--|---|
| LocalDateTimeTwoYearsAway() | Returns the DateTime value corresponding to the first day of the following year. | AddYears(LocalDateTimeTwoYearsAway(), 5) |
| LocalDateTimeYearBeforeToday() | Returns the DateTime value corresponding to the day one year ago. | AddYears(LocalDateTimeYearBeforeToday(), 5) |
| LocalDateTimeYesterday() | Returns a date-time value corresponding to Yesterday. | AddDays(LocalDateTimeYesterday(), 5) |
| Now() | Returns the current system date and time. | AddDays(Now(), 5) |
| Today() | Returns the current date. Regardless of the actual time, this function returns midnight of the current date. | AddMonths(Today(), 1) |
| UtcNow() | Returns the current system date and time, expressed as Coordinated Universal Time (UTC). | AddDays(UtcNow(), 7) |

From:
<http://www.prolin.com/dokuwiki/> - **RQest Manuals**

Permanent link:
<http://www.prolin.com/dokuwiki/expressions>

Last update: **2019/06/06 06:46**

