Numeric functions

EQL supports the following numeric functions.

Function	Description and Example
addition	The addition operator `(+)`. `SELECT NortheastSales + SoutheastSales AS EastTotalSales`
subtraction	The subtraction operator `(-)`. `SELECT SalesRevenue - TotalCosts AS Profit`
multiplication	The multiplication operator `(*)`. `SELECT Price * 0.7 AS SalePrice`
division	The division operator `(/)`. `SELECT YearTotal / 4 AS QuarterAvg`
ABS	Returns the absolute value of n. If n is `0` or a positive integer, returns n. Otherwise, n is multiplied by -1. `SELECT ABS(-1) AS one` **RESULT:** `one = 1`
CEIL	Returns the smallest integer value not less than n. `SELECT CEIL(123.45) AS x, CEIL(32) AS y, CEIL(-123.45) AS z` **RESULT:** `x = 124, y = 32, z = -123`
EXP	Exponentiation, where the base is e. Returns the value of e (the base of natural logarithms) raised to the power n. `SELECT EXP(1.0) AS baseE` **RESULT:** `baseE = e^1.0 = 2.71828182845905`
FLOOR	Returns the largest integer value not greater than n. `SELECT FLOOR(123.45) AS x, FLOOR(32) AS y, FLOOR(-123.45) AS z` **RESULT:** `x = 123, y = 32, z = -124`

Function	Description and Example
LN	Natural logarithm. Computes the logarithm of its single argument, the base of which is `e`. `SELECT LN(1.0) AS baseE` **RESULT:** `baseE = e^1.0 = 0`
LOG	Logarithm. `log(n, m)` takes two arguments, where `n` is the base, and `m` is the value you are taking the logarithm of. `Log(10,1000) = 3`
MOD	Modulo. Returns the remainder of `n` divided by `m`. `Mod(10,3) = 1` EQL uses the `fmod` floating point remainder, as defined in the C/POSIX standard.

ROUND	Returns a number rounded to the specified decimal place.
	The unary (one argument) version takes only one argument (the number to be rounded) and drops the decimal (non-integral) portion of the input. For example:
	`ROUND(8.2) returns 8`
	`ROUND(8.7) returns 9`
	The binary (two argument) version takes two arguments (the number to be rounded and a positive or negative integer that allows you to set the number of spaces at which the number is rounded). The binary version always returns a double:
	• Positive second arguments correspond to the number of places that must be returned ***after*** the decimal point. For example:
	`ROUND(123.4567, 3) returns 123.457`
	• Negative second arguments correspond to the number of places that must be returned ***before*** the decimal point. For example:
	`ROUND(123.4, -3) returns 0`
	`ROUND(1234.56, -3) returns 1000`

SIGN	Returns the sign of the argument as -1, 0, or 1, depending on whether `n` is negative, zero, or positive. The result is always a double. `SELECT SIGN(-12) AS x, SIGN(0) AS y, SIGN(12) AS z` **RESULT:** `x = -1, y = 0, z = 1`
SQRT	Returns the nonnegative square root of `n`. `SELECT SQRT(9) AS x` **RESULT:** `x = 3`

Function	Description and Example

TRUNC	Returns the number n truncated to `m` decimal places. If `m` is 0, the result has no decimal point or fractional part.
	The unary (one argument) version drops the decimal (non-integral) portion of the input. For example:
	`SELECT TRUNC(3.14159265) AS x`
	RESULT: `x = 3`
	The binary (two argument) version allows you to set the number of spaces at which the number is truncated. The binary version always returns a double. For example:
	`SELECT TRUNC(3.14159265, 3) AS y`
	RESULT: `y = 3.141`

SIN	The sine of n, where the angle of n is in radians. `SIN(3.14159/6) = 0.499999616987256`
COS	The cosine of n, where the angle of n is in radians. `COS(3.14159/3) = 0.500000766025195`
TAN	The tangent of n, where the angle of n is in radians. `TAN(3.14159/4) = 0.999998673205984`
POWER	Returns the value (as a double) of n raised to the power of `m`. `Power(2,8) = 256`
TO_DURATION	Casts a string representation of a timestamp into a number of milliseconds so that it can be used as a duration.
TO_DOUBLE	Casts a string representation of an integer as a double.
TO_INTEGER(boolean)	Casts `TRUE/FALSE` to `1/0`.

Oracle Endeca Server EQL Guide · Version 7.7.0 · January 2016