**PROPOSAL: DMN12-190**

***Original document:*** *on page 133, table 60, String Functions*

***DMN 1.2\_ballot18 clean.pdf:*** *on page 141, table 65, String Functions*

*ADD the following rows:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(parameters)** | **Parameter Domain** | **Description** | **Example** |
| split( *string, delimiter* ) | *string* is a string, *delimiter* is a pattern2 | Splits the *string* into a list of substrings, breaking at each occurrence of the *delimiter* pattern. | split( “John Doe”, “\\s” ) = [“John”, “Doe”]  split( “a;b;c;;”, “;” ) = [“a”,”b”,”c”,””,””] |

***Original document:*** *on page 134, table 61, List Functions*

***DMN 1.2\_ballot18 clean.pdf:*** *on page 142, table 66, List Functions*

*ADD the following rows:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(parameters)** | **Parameter Domain** | **Description** | **Example** |
| product( *list* )  product( *n1, …, nn*) | *list* is a list of numbers. *n1 … nn* are numbers. | Returns the product of the numbers | product( 2, 3, 4) = 24 |
| median( *list* )  median( *n1, …, nn* ) | *list* is a list of number. *n1 … nn* are numbers. | Returns the median element of the list of numbers. I.e., after sorting the list, if the list has an odd number of elements, it returns the middle element. If the list has an even number of elements, returns the average of the two middle elements. If the list is empty, returns null. | median( 8, 2, 5, 3, 4 ) = 4  median( [6, 1, 2, 3] ) = 2.5  median( [ ] ) = null |
| stddev( *list* )  stddev( *n1, …, nn* ) | *list* is a list of number. *n1 … nn* are numbers. | Returns the standard deviation of the list of numbers. If the list is empty, returns null. | stddev( 2, 4, 7, 5 ) = 2.0816659994661  stddev( [ ] ) = null |
| mode( *list* )  mode( *n1, …, nn* ) | *list* is a list of number. *n1 … nn* are numbers. | Returns the mode of the list of numbers. If the result contains multiple elements, they are returned in ascending order. If the list is empty, an empty list is returned. | mode( 6, 3, 9, 6, 6 ) = [ 6 ]  mode( [6, 1, 9, 6, 1] ) = [ 1, 6 ]  mode( [ ] ) = [ ] |

***Original document:*** *on page 135, table 62, Numeric Functions*

***DMN 1.2\_ballot18 clean.pdf:*** *on page 144, table 67, Numeric Functions*

*ADD the following rows:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name(parameters)** | **Parameter Domain** | **Description** | **Example** |
| abs( *number* ) | *number* is a number. | Returns the absolute value of the given number. | abs( 10 ) = 10  abs( -10 ) = 10 |
| modulo( *dividend*, *divisor* ) | *dividend* and *divisor* are numbers. | Returns the remainder of the division of dividend by divisor. | modulo( 12, 5 ) = 2 |
| sqrt( *number* ) | *number* is a number. | Returns the square root of the given number. If *number* is negative it returns null. | sqrt( 16 ) = 4 |
| log( *number* ) | *number* is a number | Returns the natural logarithm (base *e*) of the *number* parameter. | log( 10 ) = 2.30258509299 |
| exp( *number* ) | *number* is a number | Returns the Euler’s number *e* raised to the power of *number*. | exp( 5 ) = 148.413159102577 |
| odd( *number* ) | *number* is a number | Returns true if *number* is odd, false if it is even. | odd( 5 ) = true  odd( 2 ) = false |
| even( *number* ) | *number* is a number | Returns true if *number* is even, false if it is odd. | even( 5 ) = false  even ( 2 ) = true |