Functions that Can be applied on Numeric column

Aggregate Functions

FUNCTION	DESCRIPTION
Sum - sum(numeric_column)	Returns the sum of the 'numeric_column'
Avg - avg(numeric_column)	Returns the average of the 'numeric_column'
Min - Min(numberic_column)	Returns the minimum of the 'numeric column'
Max - max(numeric_column)	Returns the maximum of the 'numeric_column'
Count - count(column)	Returns the number of rows (values) in the given column

The above are the basic functions, we may also include statistical functions like **Mode, Median, variance**,

String functions to consider

String Functions

FUNCTION	DESCRIPTION	EXAMPLE
Concat - concat(string_column,...,string_column)	Returns the concatenated string of the given arguments. If any one of the argument is null, it returns null.	concat('abcd','ef','db') = abcdefdbd
Concat_WS - concat_ws(separator,string_column1 ,....,string_columnN)	Returns the concatenated string of the given arguments separated by the given separator. If the separator is null, it returns null.	concat_ws('-','abcd', 'ef','db') = abcd-ef-db
Insert - insert(string_column, start_pos, len, new_string)	Returns the string 'string_column', with the substring beginning at position 'start_pos' and 'len' characters long replaced by the string 'new_string'. 'start_pos' should be greater than 0. When len is zero, the 'new_string' is inserted	insert('abcddb', 3, 2, 'efgh') = abefghdb

	previous to the position 'start_pos'.	
Index of - indexof(string_column, sub_string)	Returns the index of the first occurrence of the string 'sub_string' in string 'string_column'.	indexof('abcddb','db') = 5
Left - left(string_column, len)	Returns the 'len' number of characters from the left-hand side of the string 'string_column'.	left('abcdef',3) = abc
Length - length(string_column)	Returns the character length of the string.	length('abcddb') = 6
Lowercase - lowercase(string_column)	Returns the string 'string_column' with all characters changed to lowercase.	lowercase('AbCD') = abcd
Locate - locate(sub_string, string_column, start_pos)c	Returns the index of the first occurrence of the string 'sub_string' in string 'string_column' starting at the position 'start_pos'.	locate('db',dbdb',6) = 7

Lpad - lpad(string_column, len, pad_string)	Returns the string 'string_column', left-padded to a length of 'len' characters with the string 'pad_string'. If length of the string 'string_column' is greater than 'len', then the first 'len' characters of 'string_column' is returned.	lpad('DB',5,'a') = aaaDB
Ltrim - ltrim(string_column)	Returns the string 'string_column' with leading spaces removed.	ltrim(' abcd') = abcd
Repeat - repeat(string_column,count)	repeat('Abcd',3) = 'AbcdAbcdAbcd'	
Replace - replace(string_column, from_string, to_string)	Returns the string with all occurrences of the string 'from_str' replaced by the string 'to_str'	replace('abcdac','ac','db') = abcddb
Reverse - reverse(string_column)	Returns the reverse string of 'string_column'.	reverse('abcd') = dcba

Right - right(string_column, len)	Returns the 'len' number of characters from the right-hand side of the string 'string_column'	right('abcdef',4) = cdef
Rpad - rpad(string_column, len, pad_string)	Returns the string 'string_column', right-padded to a length of 'len' characters with the string 'pad_string'. If length of the string 'string_column' is greater than 'len', then the first 'len' characters of 'string_column' is returned	rpad('DB',5,'a') = DBaaa
Rtrim - rtrim(string_column)	Returns the string 'string_column' with trailing spaces removed.	rtrim('abcd ') = abcd
Strcmp - strcmp(string_column1, string_column2)	Returns -1 if the 'string_column1' is smaller than the 'string_column2', 0 if the two strings are same, and 1 if the 'string_column1' is	strcmp('abcd', 'abcde') =-1

	greater than the 'string_column2'.	
Substring - substring(string_column, start_pos, string_len)	Returns a substring from 'string_column'. The substring begins at position 'start_pos' with the character length of 'string_len'.	substring('abcddb', 1, 4) = abcd
Trim - trim(string_column)	Returns the string with all spaces removed in prefix and suffix of the string.	trim(' abcd ') = abcd
Uppercase - uppercase(string_column)	Returns the string 'string_column' with all characters changed to uppercase.	uppercase('abcD') = ABCD

Numeric functions to discuss

FUNCTION	DESCRIPTION	EXAMPLE
Abs - abs(numeric_column)	This function returns the absolute value (number without sign) of the 'numeric_column'	pi() = 3.14159265358979
Acos - acos(numeric_column)	This function returns the arc cosine value of the specified 'numeric_column'. Returns NULL if the 'numeric_column' is not in the range-1 to 1.	pow(2,3) = 8
Asin - asin(numeric_column)	This function returns the arc sine value of the specified 'numeric_column'. Returns NULL if the 'numeric_column' is not in the range-1 to 1.	rand() = 0.28216400582 5449
Atan - atan(numeric_column)	This function returns the arc tangent value of the specified 'numeric_column'.	atan(2) = 1.107149
Atan2 - atan2(numeric_column 1, numeric_column2)	This function returns the arc tangent of the specified columns 'numeric_column1' / 'numeric_column2	atan2(0.8, 0.6) = 0.927295

Ceil - ceil(numeric_column)	This functions rounds the 'numeric_column' to the nearest integer which is greater than the 'numeric_column'	ceil(11.56) = 12
Cos - cos(numeric_column)	This function returns the cosine value of the specified 'numeric_column'	cos(0) = 1
Cot - cot(numeric_column)	This function returns the cotangent value of the specified 'numeric_column'	cot(0.25) = 3.916317
Degrees - degrees(numeric_column)	This function returns the angle in Degrees equivalent to the given Radians	degrees(1) = 57.2957795
Exp - exp(numeric_column)	This function returns the exponential value of the 'numeric_column'	exp(2) = 7.389056
Floor - floor(numeric_column)	Rounds the 'numeric_column' to the nearest integer which is less than the 'numeric_column'	floor(11.56) = 11
Greatest - greatest(numeric_colu	Gives the greatest of the given arguments.	greatest(10,20,5) = 20

mn,..., numeric_column)		
Least - least(numeric_column, ..., numeric_column)	Gives the least of the given arguments.	log10(3) = 0.477121
Ln - ln(numeric_column)	This function returns the natural logarithm of the specified 'numeric_column'	ln(5) = 1.609438
Log10 - log10(numeric_column)	This function returns the logarithm to the base-10 of the specified 'numeric_column'	log10(3) = 0.477121
Log2 - log2(numeric_column)	This function returns the logarithm to the base-2 of the 'numeric_column	log2(32) = 5
Mod - mod(numeric_column1, numeric_column2)	Returns the remainder of the 'numeric_column1' divided by 'numeric_column2'	mod(10,3) = 1
Pi - pi()	This function returns the numeric value of the pi.	pi() = 3.14159265358979

Power - pow(numeric_column1, numeric_column2)	This function returns the value of 'numeric_column1' raised to the power of 'numeric_column2'	pow(2,3) = 8
Random - rand()	Returns a random value between 0 and 1.	rand() = 0.9233482386203
Radians - radians(numeric_column)	Returns the angle in radians equivalent to the given degrees	radians(180) = 3.1415926
Round - round(numeric_column)	Returns the rounded integer value of the 'numeric_column'.	round(10.67) = 11
Sign - sign(numeric_column)	Returns-1, 0, or 1, if the 'numeric_column' is negative, zero, or positive.	sign(-23) =-1
Sin - sin(numeric_column)	Returns the sine value of the 'numeric_column'.	sin(0) = 0
Square - square(numeric_column)	Returns the square of the specified 'numeric_column'.	square(10) = 100

Square Root - sqrt(numeric_column)	Returns the square root of the specified 'numeric_column'.	sqrt(16) = 4
Tan - tan(numeric_column)	Returns the tangent value of the specified 'numeric_column'.	tan(0.5) = 0.546302

Date Functions

Date Functions		
FUNCTION	**DESCRIPTION**	**EXAMPLE**
Absolute Month - absmonth(date_column)	This function will return month and year from a given date value with the format (Month, yyyy).	absmonth('2011/8/7') = August, 2011
Absolute Quarter - absquarter(date_column)	This function will return Quarter and year from a given date value with	absquarter('2011/8/7') = Q3, 2011

	the format (Quarter, yyyy).	
Add Date - adddate(date_column, num_of_days)	This function will add the specified number of days(num_of_days) to the given date value.	adddate('2011/8/7',10) = 2011/8/17
Add Time - addtime(data_column, time)	Returns the day by adding the time to the given date column.	addtime ('2002/02/21 18:23:26', '01:20:10') = 2002/02/21 19:43:36
Current Date - currentdate()	This function will return the current date of the computer or server.	currentdate() = 15 Sep, 2011 10:06:18
Date and Time Dif - dateand timediff(Unit, From Date, To Date[optional]	This function will return the date and time difference between two date columns based on the units specified. The supported units are SECOND, MINUTE, HOUR, DAY, WEEK,	dateandtimediff(DAY, '2015-01-01', '2015-05-01')=120 The above example returns the difference between the given dates in Days.

	MONTH, QUARTER, YEAR.	
Date Diff - datediff(date_column1, date_column2)	This function will returns the difference between the two given date columns.	datediff('2011/8/11','2010/9/11') = 334
Day - day(date_column)	This function will return the day of the given date value.	day('2011/9/9') = 9
Day of Week - dayofweek(date_column)	This function will return the number of the day of week of the given date value (Sunday = 1, Monday = 2,...).	dayofweek('2011/9/9') = 6
Day of Year - dayofyear(date_column)	This function will return the number of the day of the year of the given date value (0 through 365).	dayofyear('2011/9/2') = 245
From Unixtime - fromunixtime(seconds)	This function returns the unix time for the given seconds value.	fromunixtime('1000') = 1970/01/01 05:46:40

Hour - hour(date_column)	This function returns the hour of the given date value.	hour('2011/8/7 10:35:23') = 10
Last Day - lastday(date_column)	This function will return the last day of the month for the given date value.	lastday('2011/9/7') = 2011/9/30
Make Date - makedate(year,num_of_days)	This function returns the date value for the given year and number of the day value (0 through 365)	makedate('2011','23') = 2011/1/23
Minute - minute(date_column)	This function returns the minutes of the given date value.	minute('2011/8/7 10:35:23') = 35

Modified Time -modifiedtime()	This function returns the created time of the record (if the record is newly added) or the last modified time of the record. When you apply this function, initially it will return the time at which the formula has been created. Subsequently it will return only the modified time of the record.	
Month - month(date_column)	This function returns the name of the month of the given date value.	month('2011/9/7') = September
Month Num - monthnum(date_column)	This function returns the number of the month of the given date value.	monthnum('2011/9/7') = 9
Quarter - quarter(date_column)	This function returns the quarter of the given date value.	quarter('2011/8/7') = Q3

Second - second(date_column)	This function returns the seconds of the given date/time value.	second('2011/9/7 10:35:23') = 23
Sub Date - subdate(date_column,num_of_days)	This function returns the date by subtracting the number of days(num_of_days) from the given date value.	subdate('2011/9/15','6') = 1990/9/9
Sub Time - subtime(date_column,time)	This function returns the date by subtracting the time from the given date with time value.	subtime('2011/02/21 18:23:26','01:20:10') = 2011/02/21 17:03:16
Week Day - weekday(date_column)	This function returns weekday name (Sunday, Monday...) of the given date value.	weekday('2011/9/14') = Wednesday
Year - year(date_column)	This function returns year from the given date value.	year('2011/9/7') = 2011