Apache Hive 内置函数(Builtin Function)列表

本文所列的 Hive 函数均为 Hive 内置的，共计294个，Hive 版本为 3.1.0。

如果想及时了
解Spark、Hadoop或者Hbase相关的文章，欢迎关注微信公共帐号：iteblog_hadoop

!

! a - Logical not，和not逻辑操作符含义一致

```
hive> select !(true);
OK
false
```

!=

a != b - Returns TRUE if a is not equal to b，和操作符含义一致

```
hive> select 1 <> 2;
```

OK
true

$sum0

$sum0(x) - 返回一组数字的总和, 如果没有数字为空范围0

```
hive> select `$sum0`(1L);
OK
1
```

%

a % b - Returns the remainder when dividing a by b

&

a & b - Bitwise and

```
hive> SELECT 3 & 5 FROM src LIMIT 1;
1
```

*

a * b - Multiplies a by b

+

a + b - Returns a+b

-

a - b - Returns the difference a-b

/

a / b - Divide a by b

> SELECT 3 / 2 FROM src LIMIT 1;
1.5

<

a < b - Returns TRUE if a is less than b

<=

a <= b - Returns TRUE if a is not greater than b

<=>

a <=> b - Returns same result with EQUAL(=) operator for non-null operands, but returns TRUE if both are NULL, FALSE if one of the them is NULL

<>

a <> b - Returns TRUE if a is not equal to b，和!=含义一致

=

a = b - Returns TRUE if a equals b and false otherwise，和==含义一致

==

a == b - Returns TRUE if a equals b and false otherwise，和=含义一致

>

a > b - Returns TRUE if a is greater than b

>=

a >= b - Returns TRUE if a is not smaller than b

^

a ^ b - Bitwise exclusive or

> SELECT 3 ^ 5 FROM src LIMIT 1;
2

abs

abs(x) - returns the absolute value of x

> SELECT abs(0) FROM src LIMIT 1;
0
> SELECT abs(-5) FROM src LIMIT 1;
5

acos

acos(x) - returns the arc cosine of x if -1<=x<=1 or NULL otherwise

> SELECT acos(1) FROM src LIMIT 1;
0
> SELECT acos(2) FROM src LIMIT 1;
NULL

add_months

add_months(start_date, num_months) - Returns the date that is num_months after start_date. start_date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. num_months is a number. The time part of start_date is ignored.

> SELECT add_months('2009-08-31', 1) FROM src LIMIT 1;
'2009-09-30'

aes_decrypt

aes_decrypt(input binary, key string/binary) - Decrypt input using AES.
AES (Advanced Encryption Standard) algorithm. Key lengths of 128, 192 or 256 bits can be used. 192 and 256 bits keys can be used if Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files are installed. If either argument is NULL or the key length is

not one of the permitted values, the return value is NULL.

> SELECT aes_decrypt(unbase64('y6Ss+zCYObpCbgfWfyNWTw=='), '1234567890123456');
'ABC'

aes_encrypt

aes_encrypt(input string/binary, key string/binary) - Encrypt input using AES.
AES (Advanced Encryption Standard) algorithm. Key lengths of 128, 192 or 256 bits can be used. 192 and 256 bits keys can be used if Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files are installed. If either argument is NULL or the key length is not one of the permitted values, the return value is NULL.

> SELECT base64(aes_encrypt('ABC', '1234567890123456'));
 'y6Ss+zCYObpCbgfWfyNWTw=='

and

a1 and a2 and ... and an - Logical and

array

array(n0, n1...) - Creates an array with the given elements

array_contains

array_contains(array, value) - Returns TRUE if the array contains value.

> SELECT array_contains(array(1, 2, 3), 2) FROM src LIMIT 1;
true

ascii

ascii(str) - returns the numeric value of the first character of str
Returns 0 if str is empty or NULL if str is NULL

> SELECT ascii('222') FROM src LIMIT 1; 50

```
> SELECT ascii(2) FROM src LIMIT 1;
50
```

asin

asin(x) - returns the arc sine of x if -1<=x<=1 or NULL otherwise

```
> SELECT asin(0) FROM src LIMIT 1;
0
> SELECT asin(2) FROM src LIMIT 1;
NULL
```

assert_true

assert_true(condition) - Throw an exception if 'condition' is not true.

```
 > SELECT assert_true(x >= 0) FROM src LIMIT 1;
NULL
```

assert_true_oom

assert_true_oom - assertation failed; Simulated OOM

```
select assert_true_oom(${hiveconf:zzz}>sum(1)) from tu join tv on (tu.id_uv=tv.id_uv) where u<
10 and v>1
```

atan

atan(x) - returns the atan (arctan) of x (x is in radians)

```
> SELECT atan(0) FROM src LIMIT 1;
0
```

avg

avg(x) - Returns the mean of a set of numbers

base64

base64(bin) - Convert the argument from binary to a base 64 string

between

between a [NOT] BETWEEN b AND c - evaluate if a is [not] in between b and c

bin

bin(n) - returns n in binary
n is a BIGINT. Returns NULL if n is NULL.

> SELECT bin(13) FROM src LIMIT 1
'1101'

bloom_filter

Generic UDF to generate Bloom Filter

bround

bround(x[, d]) - round x to d decimal places using HALF_EVEN rounding mode.
Banker's rounding. The value is rounded to the nearest even number. Also known as Gaussian rounding.

> SELECT bround(12.25, 1);
12.2

cardinality_violation

cardinality_violation(n0, n1...) - raises Cardinality Violation

case

CASE a WHEN b THEN c [WHEN d THEN e]* [ELSE f] END - When a = b, returns c; when a = d, return e; else return f

```
SELECT
CASE deptno
  WHEN 1 THEN Engineering
  WHEN 2 THEN Finance
  ELSE admin
END,
CASE zone
  WHEN 7 THEN Americas
  ELSE Asia-Pac
END
FROM emp_details
```

cbrt

cbrt(double) - Returns the cube root of a double value.

```
> SELECT cbrt(27.0);
3.0
```

ceil

ceil(x) - Find the smallest integer not smaller than x
Synonyms: ceiling

```
> SELECT ceil(-0.1) FROM src LIMIT 1;
0
> SELECT ceil(5) FROM src LIMIT 1;
5
```

ceiling

ceiling(x) - Find the smallest integer not smaller than x
Synonyms: ceil

```
> SELECT ceiling(-0.1) FROM src LIMIT 1;
0
> SELECT ceiling(5) FROM src LIMIT 1;
5
```

char_length

char_length(str | binary) - Returns the number of characters in str or binary data

character_length

character_length(str | binary) - Returns the number of characters in str or binary data

chr

chr(str) - convert n where n : [0, 256) into the ascii equivalent as a varchar.If n is less than 0 return the empty string. If n > 256, return chr(n % 256).

```
> SELECT chr('48') FROM src LIMIT 1;
'0'
> SELECT chr('65') FROM src LIMIT 1;
'A'
```

coalesce

coalesce(a1, a2, ...) - Returns the first non-null argument

```
> SELECT coalesce(NULL, 1, NULL) FROM src LIMIT 1;
1
```

collect_list

collect_list(x) - Returns a list of objects with duplicates

collect_set

collect_set(x) - Returns a set of objects with duplicate elements eliminated

compute_stats

compute_stats(x) - Returns the statistical summary of a set of primitive type values.

concat

concat(str1, str2, ... strN) - returns the concatenation of str1, str2, ... strN or concat(bin1, bin2, ... binN) - returns the concatenation of bytes in binary data bin1, bin2, ... binN
Returns NULL if any argument is NULL.

> SELECT concat('abc', 'def') FROM src LIMIT 1;
'abcdef'

concat_ws

concat_ws(separator, [string | array(string)]+) - returns the concatenation of the strings separated by the separator.

> SELECT concat_ws('.', 'www', array('iteblog', 'com')) FROM src LIMIT 1;
'www.iteblog.com'

context_ngrams

context_ngrams(expr, array<string1 , string2, ...>, k, pf) estimates the top-k most frequent n-grams that fit into the specified context. The second parameter specifies a string of words that specify the positions of the n-gram elements, with a null value standing in for a 'blank' that must be filled by an n-gram element.
The primary expression must be an array of strings, or an array of arrays of strings, such as the return type of the sentences() UDF. The second parameter specifies the context -- for example, array("i", "love", null) -- which would estimate the top 'k' words that follow the phrase "i love" in the primary expression. The optional fourth parameter 'pf' controls the memory used by the heuristic. Larger values will yield better accuracy, but use more memory.
Example usage:

SELECT context_ngrams(sentences(lower(review)), array("i", "love", null, null), 10) FROM movi
es

would attempt to determine the 10 most common two-word phrases that follow "i love" in a database of free-form natural language movie reviews.

conv

conv(num, from_base, to_base) - convert num from from_base to to_base
If to_base is negative, treat num as a signed integer,otherwise, treat it as an unsigned integer.

> SELECT conv('100', 2, 10) FROM src LIMIT 1;
'4'
> SELECT conv(-10, 16, -10) FROM src LIMIT 1;
'16'

corr

corr(x,y) - Returns the Pearson coefficient of correlation between a set of number pairs

The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored. If the function is applied to an empty set or
a singleton set, NULL will be returned. Otherwise, it computes the following:
  COVAR_POP(x,y)/(STDDEV_POP(x)*STDDEV_POP(y))
where neither x nor y is null,
COVAR_POP is the population covariance,
and STDDEV_POP is the population standard deviation.

cos

cos(x) - returns the cosine of x (x is in radians)

 > SELECT cos(0) FROM src LIMIT 1;
1

count

count(*) - Returns the total number of retrieved rows, including rows containing NULL values.
count(expr) - Returns the number of rows for which the supplied expression is non-NULL.
count(DISTINCT expr[, expr...]) - Returns the number of rows for which the supplied

expression(s) are unique and non-NULL.

covar_pop

covar_pop(x,y) - Returns the population covariance of a set of number pairs
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored. If the function is applied to an empty set, NULL
will be returned. Otherwise, it computes the following:

```
  (SUM(x*y)-SUM(x)*SUM(y)/COUNT(x,y))/COUNT(x,y)
```

where neither x nor y is null.

covar_samp

covar_samp(x,y) - Returns the sample covariance of a set of number pairs
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored. If the function is applied to an empty set, NULL
will be returned. Otherwise, it computes the following:

```
  (SUM(x*y)-SUM(x)*SUM(y)/COUNT(x,y))/(COUNT(x,y)-1)
```

where neither x nor y is null.

crc32

crc32(str or bin) - Computes a cyclic redundancy check value for string or binary argument and returns bigint value.

```
 > SELECT crc32('ABC');
 2743272264
 > SELECT crc32(binary('ABC'));
 2743272264
```

create_union

create_union(tag, obj1, obj2, obj3, ...) - Creates a union with the object for given tag

> SELECT create_union(1, 1, "one") FROM src LIMIT 1;
{1:"one"}

cume_dist

cume_dist - The CUME_DIST function (defined as the inverse of percentile in some statistical books) computes the position of a specified value relative to a set of values. To compute the CUME_DIST of a value x in a set S of size N, you use the formula: CUME_DIST(x) = number of values in S coming before and including x in the specified order/ N

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFCumeDist
Function type:BUILTIN

current_authorizer

current_authorizer() - Returns the current authorizer (class name of the authorizer).
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFCurrentAuthorizer
Function type:BUILTIN

current_database

current_database() - returns currently using database name

current_date

返回系统当前日期

hive> select current_date();
OK
2017-02-22

current_groups

current_groups() - Returns all groups the current user belongs to.

current_timestamp

返回系统当前时间

```
hive> select current_timestamp();
OK
2017-02-22 14:04:33.332
```

current_user

current_user() - 返回当前用户名称，SessionState UserFromAuthenticator

```
hive> select current_user();
OK
iteblog
```

date_add

date_add(start_date, num_days) - Returns the date that is num_days after start_date. start_date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. num_days is a number. The time part of start_date is ignored.

```
  > SELECT date_add('2009-07-30', 1) FROM src LIMIT 1;
 '2009-07-31'
```

date_format

date_format(date/timestamp/string, fmt) - converts a date/timestamp/string to a value of string in the format specified by the date format fmt.
Supported formats are SimpleDateFormat formats - https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html. Second argument fmt should be constant.

```
> SELECT date_format('2015-04-08', 'y');
 '2015'
```

date_sub

date_sub(start_date, num_days) - Returns the date that is num_days before start_date.
start_date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. num_days is a number. The time part of start_date is ignored.

```
  > SELECT date_sub('2009-07-30', 1) FROM src LIMIT 1;
 '2009-07-29'
```

datediff

datediff(date1, date2) - Returns the number of days between date1 and date2
date1 and date2 are strings in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. The time parts are ignored.If date1 is earlier than date2, the result is negative.

```
> SELECT datediff('2009-07-30', '2009-07-31') FROM src LIMIT 1;
1
```

day

day(param) - Returns the day of the month of date/timestamp, or day component of interval
Synonyms: dayofmonth
param can be one of:

- A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'.
- A date value
- A timestamp value
- A day-time interval value

```
> SELECT day('2009-07-30') FROM src LIMIT 1;
30
```

dayofmonth

dayofmonth(param) - Returns the day of the month of date/timestamp, or day component of interval
Synonyms: day

param can be one of:

- A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'.
- A date value
- A timestamp value
- A day-time interval value

> SELECT dayofmonth('2009-07-30') FROM src LIMIT 1;
30

dayofweek

dayofweek(param) - Returns the day of the week of date/timestamp (1 = Sunday, 2 = Monday, ..., 7 = Saturday)
param can be one of:

- A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'.
- A date value
- A timestamp valueExample:

  > SELECT dayofweek('2009-07-30') FROM src LIMIT 1;
  5

Function class:org.apache.hadoop.hive.ql.udf.UDFDayOfWeek
Function type:BUILTIN

decode

decode(bin, str) - Decode the first argument using the second argument character set
Possible options for the character set are 'US-ASCII', 'ISO-8859-1',
'UTF-8', 'UTF-16BE', 'UTF-16LE', and 'UTF-16'. If either argument
is null, the result will also be null

degrees

degrees(x) - Converts radians to degrees

> SELECT degrees(30) FROM src LIMIT 1;
-1

dense_rank

There is no documentation for function 'dense_rank'

div

a div b - Divide a by b rounded to the long integer

```
> SELECT 3 div 2 FROM src LIMIT 1;
1
```

e

e() - returns E

```
> SELECT e() FROM src LIMIT 1;
2.718281828459045
```

elt

elt(n, str1, str2, ...) - returns the n-th string

```
> SELECT elt(1, 'face', 'book') FROM src LIMIT 1;
'face'
```

encode

encode(str, str) - Encode the first argument using the second argument character set
Possible options for the character set are 'US-ASCII', 'ISO-8859-1',
'UTF-8', 'UTF-16BE', 'UTF-16LE', and 'UTF-16'. If either argument
is null, the result will also be null

enforce_constraint

enforce_constraint(x) - Internal UDF to enforce CHECK and NOT NULL constraint
For internal use only
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFEnforceConstraint

Function type:BUILTIN

ewah_bitmap

ewah_bitmap(expr) - Returns an EWAH-compressed bitmap representation of a column.

ewah_bitmap_and

ewah_bitmap_and(b1, b2) - Return an EWAH-compressed bitmap that is the bitwise AND of two bitmaps.

ewah_bitmap_empty

ewah_bitmap_empty(bitmap) - Predicate that tests whether an EWAH-compressed bitmap is all zeros

ewah_bitmap_or

ewah_bitmap_or(b1, b2) - Return an EWAH-compressed bitmap that is the bitwise OR of two bitmaps.

exp

exp(x) - Returns e to the power of x

```
> SELECT exp(0) FROM src LIMIT 1;
1
```

explode

explode(a) - separates the elements of array a into multiple rows, or the elements of a map into multiple rows and columns

factorial

factorial(int) - Returns n factorial. Valid n is [0..20].
Returns null if n is out of [0..20] range.

```
> SELECT factorial(5);
120
```

extract_union

extract_union(union[, tag]) - Recursively explodes unions into structs or simply extracts the given tag.

```
 > SELECT extract_union({0:"foo"}).tag_0 FROM src;
 foo
 > SELECT extract_union({0:"foo"}).tag_1 FROM src;
 null
 > SELECT extract_union({0:"foo"}, 0) FROM src;
 foo
 > SELECT extract_union({0:"foo"}, 1) FROM src;
 null
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFExtractUnion
Function type:BUILTIN

factorial

factorial(int) - Returns n factorial. Valid n is [0..20].
Returns null if n is out of [0..20] range.
Example:

```
 > SELECT factorial(5);
 120
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFFactorial
Function type:BUILTIN

field

field(str, str1, str2, ...) - returns the index of str in the str1,str2,... list or 0 if not found
All primitive types are supported, arguments are compared using str.equals(x). If str is NULL, the return value is 0.

find_in_set

find_in_set(str,str_array) - Returns the first occurrence of str in str_array where str_array is a comma-delimited string. Returns null if either argument is null. Returns 0 if the first argument

has any commas.

```
> SELECT find_in_set('ab','abc,b,ab,c,def') FROM src LIMIT 1;
3
> SELECT * FROM src1 WHERE NOT find_in_set(key,'311,128,345,956') = 0;
311  val_311
128
```

first_value

first_value - This takes at most two parameters. The first parameter is the column for which you want the first value, the second (optional) parameter must be a boolean which is false by default. If set to true it skips null values.

floor

floor(x) - Find the largest integer not greater than x

```
> SELECT floor(-0.1) FROM src LIMIT 1;
-1
> SELECT floor(5) FROM src LIMIT 1;
5
```

floor_day

floor_day(param) - Returns the timestamp at a day granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_day(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-dd 00:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorDay
Function type:BUILTIN

floor_hour

floor_hour(param) - Returns the timestamp at a hour granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_hour(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-dd HH:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorHour
Function type:BUILTIN

floor_minute

floor_minute(param) - Returns the timestamp at a minute granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_minute(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-dd HH:mm:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorMinute
Function type:BUILTIN

floor_month

floor_month(param) - Returns the timestamp at a month granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_month(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-01 00:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorMonth
Function type:BUILTIN

floor_quarter

floor_quarter(param) - Returns the timestamp at a quarter granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_quarter(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-xx-01 00:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorQuarter
Function type:BUILTIN

floor_second

floor_second(param) - Returns the timestamp at a second granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_second(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-dd HH:mm:ss
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorSecond
Function type:BUILTIN

floor_week

floor_week(param) - Returns the timestamp at a week granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_week(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-MM-xx 00:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorWeek
Function type:BUILTIN

floor_year

floor_year(param) - Returns the timestamp at a year granularity
param needs to be a timestamp value
Example:

```
  > SELECT floor_year(CAST('yyyy-MM-dd HH:mm:ss' AS TIMESTAMP)) FROM src;
 yyyy-01-01 00:00:00
```

Function class:org.apache.hadoop.hive.ql.udf.UDFDateFloorYear
Function type:BUILTIN

format_number

format_number(X, D or F) - Formats the number X to a format like '#,###,###.##', rounded to D decimal places, Or Uses the format specified F to format, and returns the result as a string. If D is 0, the result has no decimal point or fractional part. This is supposed to function like MySQL's FORMAT

```
> SELECT format_number(12332.123456, 4) FROM src LIMIT 1;
'12,332.1235'
> SELECT format_number(12332.123456, '####################.###') FROM src LIMIT 1;
'12332.123'
```

from_unixtime

from_unixtime(unix_time, format) - returns unix_time in the specified format

```
> SELECT from_unixtime(0, 'yyyy-MM-dd HH:mm:ss') FROM src LIMIT 1;
'1970-01-01 00:00:00'
```

from_utc_timestamp

from_utc_timestamp(timestamp, string timezone) - Assumes given timestamp is UTC and converts to given timezone (as of Hive 0.8.0)

get_json_object

get_json_object(json_txt, path) - Extract a json object from path

Extract json object from a json string based on json path specified, and return json string of th
e extracted json object. It will return null if the input json string is invalid.
A limited version of JSONPath supported:
  $: Root object
  .   : Child operator
  [] : Subscript operator for array
  *   : Wildcard for []
Syntax not supported that's worth noticing:
  '' : Zero length string as key
  .. : Recursive descent
  @   : Current object/element
  () : Script expression
  ?() : Filter (script) expression.
  [,] : Union operator
  [start:end:step] : array slice operator

get_splits

get_splits(string,int) - Returns an array of length int serialized splits for the referenced tables string.

greatest

greatest(v1, v2, ...) - Returns the greatest value in a list of values

> SELECT greatest(2, 3, 1) FROM src LIMIT 1;
3

grouping

grouping(a, p1, ..., pn) - Indicates whether a specified column expression in is aggregated or not. Returns 1 for aggregated or 0 for not aggregated.
a is the grouping id, p1...pn are the indices we want to extract
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFGrouping
Function type:BUILTIN

hash

hash(a1, a2, ...) - Returns a hash value of the arguments
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFHash

Function type:BUILTIN

hex

hex(n, bin, or str) - Convert the argument to hexadecimal
If the argument is a string, returns two hex digits for each character in the string.
If the argument is a number or binary, returns the hexadecimal representation.
Example:

```
 > SELECT hex(17) FROM src LIMIT 1;
 'H1'
 > SELECT hex('Facebook') FROM src LIMIT 1;
 '46616365626F6F6B'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFHex
Function type:BUILTIN

histogram_numeric

histogram_numeric(expr, nb) - Computes a histogram on numeric 'expr' using nb bins.
Example:

```
> SELECT histogram_numeric(val, 3) FROM src;
[{"x":100,"y":14.0},{"x":200,"y":22.0},{"x":290.5,"y":11.0}]
```

The return value is an array of (x,y) pairs representing the centers of the histogram's bins. As the value of 'nb' is increased, the histogram approximationgets finer-grained, but may yield artifacts around outliers. In practice, 20-40 histogram bins appear to work well, with more bins being required for skewed or smaller datasets. Note that this function creates a histogram with non-uniform bin widths. It offers no guarantees in terms of the mean-squared-error of the histogram, but in practice is comparable to the histograms produced by the R/S-Plusstatistical computing packages.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFHistogramNumeric
Function type:BUILTIN

hour

hour(param) - Returns the hour componemnt of the string/timestamp/interval
param can be one of:

1. A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'HH:mm:ss'.
2. A timestamp value
3. A day-time interval valueExample:

```
  > SELECT hour('2009-07-30 12:58:59') FROM src LIMIT 1;
 12
 > SELECT hour('12:58:59') FROM src LIMIT 1;
 12
```

Function class:org.apache.hadoop.hive.ql.udf.UDFHour
Function type:BUILTIN

if

IF(expr1,expr2,expr3) - If expr1 is TRUE (expr1 0 and expr1 NULL) then IF() returns expr2; otherwise it returns expr3. IF() returns a numeric or string value, depending on the context in which it is used.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFIf
Function type:BUILTIN

in

test in(val1, val2...) - returns true if test equals any valN
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFIn
Function type:BUILTIN

in_bloom_filter

in_bloom_filter - GenericUDF to lookup a value in BloomFilter
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFInBloomFilter
Function type:BUILTIN

in_file

in_file(str, filename) - Returns true if str appears in the file
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFInFile
Function type:BUILTIN

index

index(a, n) - Returns the n-th element of a
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFIndex

Function type:BUILTIN

initcap

initcap(str) - Returns str, with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space.
Example:

```
 > SELECT initcap('tHe soap') FROM src LIMIT 1;
 'The Soap'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFInitCap
Function type:BUILTIN

inline

inline(ARRAY(STRUCT()[,STRUCT()] - explodes and array and struct into a table
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFInline
Function type:BUILTIN

instr

instr(str, substr) - Returns the index of the first occurance of substr in str
Example:

```
 > SELECT instr('Facebook', 'boo') FROM src LIMIT 1;
 5
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFInstr
Function type:BUILTIN

internal_interval

internal_interval(intervalType,intervalArg)
this method is not designed to be used by directly calling it - it provides internal support for 'INTERVAL (intervalArg) intervalType' constructs
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFInternalInterval
Function type:BUILTIN

isfalse

isfalse a - Returns true if a is FALSE and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPFalse
Function type:BUILTIN

isnotfalse

isnotfalse a - Returns true if a is NOT FALSE and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNotFalse
Function type:BUILTIN

isnotnull

isnotnull a - Returns true if a is not NULL and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNotNull
Function type:BUILTIN

isnottrue

isnottrue a - Returns true if a is NOT TRUE and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNotTrue
Function type:BUILTIN

isnull

isnull a - Returns true if a is NULL and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNull
Function type:BUILTIN

istrue

istrue a - Returns true if a is TRUE and false otherwise
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPTrue
Function type:BUILTIN

java_method

java_method(class,method[,arg1[,arg2..]]) calls method with reflection
Synonyms: reflect
Use this UDF to call Java methods by matching the argument signature

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFReflect
Function type:BUILTIN

json_tuple

json_tuple(jsonStr, p1, p2, ..., pn) - like get_json_object, but it takes multiple names and return a tuple. All the input parameters and output column types are string.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFJSONTuple
Function type:BUILTIN

lag

LAG (scalar_expression [,offset] [,default]) OVER ([query_partition_clause] order_by_clause); The LAG function is used to access data from a previous row.
Example:

```
 select p1.p_mfgr, p1.p_name, p1.p_size,
 p1.p_size - lag(p1.p_size,1,p1.p_size) over( distribute by p1.p_mfgr sort by p1.p_name) as delta
Sz
 from part p1 join part p2 on p1.p_partkey = p2.p_partkey
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLag
Function type:BUILTIN

last_day

last_day(date) - Returns the last day of the month which the date belongs to.
date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. The time part of date is ignored.
Example:

```
  > SELECT last_day('2009-01-12') FROM src LIMIT 1;
 '2009-01-31'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLastDay
Function type:BUILTIN

last_value

last_value - This takes at most two parameters. The first parameter is the column for which you want the last value, the second (optional) parameter must be a boolean which is false by default. If set to true it skips null values.

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFLastValue
Function type:BUILTIN

lcase

lcase(str) - Returns str with all characters changed to lowercase
Synonyms: lower
Example:

```
 > SELECT lcase('Facebook') FROM src LIMIT 1;
 'facebook'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLower
Function type:BUILTIN

lead

LEAD (scalar_expression [,offset] [,default]) OVER ([query_partition_clause] order_by_clause);
The LEAD function is used to return data from the next row.
Example:

```
select p_name, p_retailprice, lead(p_retailprice) over() as l1,
lag(p_retailprice) over() as l2
from part
where p_retailprice = 1173.15
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLead
Function type:BUILTIN

least

least(v1, v2, ...) - Returns the least value in a list of values
Example:

```
 > SELECT least(2, 3, 1) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLeast
Function type:BUILTIN

length

length(str | binary) - Returns the length of str or number of bytes in binary data
Example:

```
 > SELECT length('Facebook') FROM src LIMIT 1;
 8
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLength
Function type:BUILTIN

levenshtein

levenshtein(str1, str2) - This function calculates the Levenshtein distance between two strings. Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. It is named after Vladimir Levenshtein, who considered this distance in 1965.Example:

```
 > SELECT levenshtein('kitten', 'sitting');
3
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLevenshtein
Function type:BUILTIN

like

like(str, pattern) - Checks if str matches pattern
Example:

```
 > SELECT a.* FROM srcpart a WHERE a.hr like '%2' LIMIT 1;
 27      val_27  2008-04-08      12
```

Function class:org.apache.hadoop.hive.ql.udf.UDFLike
Function type:BUILTIN

likeall

test likeall(pattern1, pattern2...) - returns true if test matches all patterns patternN. Returns NULL if the expression on the left hand side is NULL or if one of the patterns in the list is NULL.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLikeAll
Function type:BUILTIN

likeany

test likeany(pattern1, pattern2...) - returns true if test matches any patterns patternN. Returns NULL if the expression on the left hand side is NULL or if one of the patterns in the list is NULL.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLikeAny
Function type:BUILTIN

ln

ln(x) - Returns the natural logarithm of x
Example:

```
 > SELECT ln(1) FROM src LIMIT 1;
 0
```

Function class:org.apache.hadoop.hive.ql.udf.UDFLn
Function type:BUILTIN

locate

locate(substr, str[, pos]) - Returns the position of the first occurance of substr in str after position pos
Example:

```
 > SELECT locate('bar', 'foobarbar', 5) FROM src LIMIT 1;
 7
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLocate
Function type:BUILTIN

log

log([b], x) - Returns the logarithm of x with base b
Example:

```
 > SELECT log(13, 13) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFLog
Function type:BUILTIN

log10

log10(x) - Returns the logarithm of x with base 10
Example:

```
 > SELECT log10(10) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFLog10
Function type:BUILTIN

log2

log2(x) - Returns the logarithm of x with base 2
Example:

```
 > SELECT log2(2) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFLog2
Function type:BUILTIN

logged_in_user

logged_in_user() - Returns logged in user name
SessionState GetUserName - the username provided at session initialization
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLoggedInUser
Function type:BUILTIN

lower

lower(str) - Returns str with all characters changed to lowercase
Synonyms: lcase
Example:

```
> SELECT lower('Facebook') FROM src LIMIT 1;
'facebook'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLower
Function type:BUILTIN

lpad

lpad(str, len, pad) - Returns str, left-padded with pad to a length of len
If str is longer than len, the return value is shortened to len characters.
In case of empty pad string, the return value is null.
Example:

```
> SELECT lpad('hi', 5, '??') FROM src LIMIT 1;
'???hi'
> SELECT lpad('hi', 1, '??') FROM src LIMIT 1;
'h'
> SELECT lpad('hi', 5, '') FROM src LIMIT 1;
null
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLpad
Function type:BUILTIN

ltrim

ltrim(str) - Removes the leading space characters from str

Example:

```
 > SELECT ltrim('   facebook') FROM src LIMIT 1;
 'facebook'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFLTrim
Function type:BUILTIN

map

map(key0, value0, key1, value1...) - Creates a map with the given key/value pairs
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMap
Function type:BUILTIN

map_keys

map_keys(map) - Returns an unordered array containing the keys of the input map.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMapKeys
Function type:BUILTIN

map_values

map_values(map) - Returns an unordered array containing the values of the input map.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMapValues
Function type:BUILTIN

mask

masks the given value
Examples:
mask(ccn)
mask(ccn, 'X', 'x', '0')
mask(ccn, 'x', 'x', 'x')
Arguments:
mask(value, upperChar, lowerChar, digitChar, otherChar, numberChar, dayValue, monthValue, yearValue)
value - value to mask. Supported types: TINYINT, SMALLINT, INT, BIGINT, STRING, VARCHAR, CHAR, DATE
upperChar - character to replace upper-case characters with. Specify -1 to retain original character. Default value: 'X'
lowerChar - character to replace lower-case characters with. Specify -1 to retain original character. Default value: 'x'

digitChar - character to replace digit characters with. Specify -1 to retain original character. Default value: 'n'
otherChar - character to replace all other characters with. Specify -1 to retain original character. Default value: -1
numberChar - character to replace digits in a number with. Valid values: 0-9. Default value: '1'
dayValue - value to replace day field in a date with. Specify -1 to retain original value. Valid values: 1-31. Default value: 1
monthValue - value to replace month field in a date with. Specify -1 to retain original value. Valid values: 0-11. Default value: 0
yearValue - value to replace year field in a date with. Specify -1 to retain original value. Default value: 0

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMask
Function type:BUILTIN

mask_first_n

masks the first n characters of the value
Examples:
mask_first_n(ccn, 8)
mask_first_n(ccn, 8, 'x', 'x', 'x')
Arguments:
mask(value, charCount, upperChar, lowerChar, digitChar, otherChar, numberChar)
value - value to mask. Supported types: TINYINT, SMALLINT, INT, BIGINT, STRING, VARCHAR, CHAR
charCount - number of characters. Default value: 4
upperChar - character to replace upper-case characters with. Specify -1 to retain original character. Default value: 'X'
lowerChar - character to replace lower-case characters with. Specify -1 to retain original character. Default value: 'x'
digitChar - character to replace digit characters with. Specify -1 to retain original character. Default value: 'n'
otherChar - character to replace all other characters with. Specify -1 to retain original character. Default value: -1
numberChar - character to replace digits in a number with. Valid values: 0-9. Default value: '1'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMaskFirstN
Function type:BUILTIN

mask_hash

returns hash of the given value
Examples:
mask_hash(value)
Arguments:

value - value to mask. Supported types: STRING, VARCHAR, CHAR
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMaskHash
Function type:BUILTIN

mask_last_n

masks the last n characters of the value
Examples:
mask_last_n(ccn, 8)
mask_last_n(ccn, 8, 'x', 'x', 'x')
Arguments:
mask_last_n(value, charCount, upperChar, lowerChar, digitChar, otherChar, numberChar)
value - value to mask. Supported types: TINYINT, SMALLINT, INT, BIGINT, STRING, VARCHAR, CHAR
charCount - number of characters. Default value: 4
upperChar - character to replace upper-case characters with. Specify -1 to retain original character. Default value: 'X'
lowerChar - character to replace lower-case characters with. Specify -1 to retain original character. Default value: 'x'
digitChar - character to replace digit characters with. Specify -1 to retain original character. Default value: 'n'
otherChar - character to replace all other characters with. Specify -1 to retain original character. Default value: -1
numberChar - character to replace digits in a number with. Valid values: 0-9. Default value: '1'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMaskLastN
Function type:BUILTIN

mask_show_first_n

masks all but first n characters of the value
Examples:
mask_show_first_n(ccn, 8)
mask_show_first_n(ccn, 8, 'x', 'x', 'x')
Arguments:
mask_show_first_n(value, charCount, upperChar, lowerChar, digitChar, otherChar, numberChar)
value - value to mask. Supported types: TINYINT, SMALLINT, INT, BIGINT, STRING, VARCHAR, CHAR
charCount - number of characters. Default value: 4
upperChar - character to replace upper-case characters with. Specify -1 to retain original character. Default value: 'X'
lowerChar - character to replace lower-case characters with. Specify -1 to retain original character. Default value: 'x'
digitChar - character to replace digit characters with. Specify -1 to retain original character.

Default value: 'n'
otherChar - character to replace all other characters with. Specify -1 to retain original character. Default value: -1
numberChar - character to replace digits in a number with. Valid values: 0-9. Default value: '1'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMaskShowFirstN
Function type:BUILTIN

mask_show_last_n

masks all but last n characters of the value
Examples:
mask_show_last_n(ccn, 8)
mask_show_last_n(ccn, 8, 'x', 'x', 'x')
Arguments:
mask_show_last_n(value, charCount, upperChar, lowerChar, digitChar, otherChar, numberChar)
value - value to mask. Supported types: TINYINT, SMALLINT, INT, BIGINT, STRING, VARCHAR, CHAR
charCount - number of characters. Default value: 4
upperChar - character to replace upper-case characters with. Specify -1 to retain original character. Default value: 'X'
lowerChar - character to replace lower-case characters with. Specify -1 to retain original character. Default value: 'x'
digitChar - character to replace digit characters with. Specify -1 to retain original character. Default value: 'n'
otherChar - character to replace all other characters with. Specify -1 to retain original character. Default value: -1
numberChar - character to replace digits in a number with. Valid values: 0-9. Default value: '1'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMaskShowLastN
Function type:BUILTIN

matchpath

There is no documentation for function 'matchpath'
Function class:org.apache.hadoop.hive.ql.udf.ptf.MatchPath$MatchPathResolver
Function type:BUILTIN

max

max(expr) - Returns the maximum value of expr
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFMax
Function type:BUILTIN

md5

md5(str or bin) - Calculates an MD5 128-bit checksum for the string or binary.
The value is returned as a string of 32 hex digits, or NULL if the argument was NULL.
Example:

```
 > SELECT md5('ABC');
 '902fbdd2b1df0c4f70b4a5d23525e932'
 > SELECT md5(binary('ABC'));
 '902fbdd2b1df0c4f70b4a5d23525e932'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFMd5
Function type:BUILTIN

min

min(expr) - Returns the minimum value of expr
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFMin
Function type:BUILTIN

minute

minute(param) - Returns the minute component of the string/timestamp/interval
param can be one of:
1. A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'HH:mm:ss'.
2. A timestamp value
3. A day-time interval valueExample:

```
  > SELECT minute('2009-07-30 12:58:59') FROM src LIMIT 1;
 58
 > SELECT minute('12:58:59') FROM src LIMIT 1;
 58
```

Function class:org.apache.hadoop.hive.ql.udf.UDFMinute
Function type:BUILTIN

mod

a mod b - Returns the remainder when dividing a by b

Synonyms: %
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPMod
Function type:BUILTIN

month

month(param) - Returns the month component of the date/timestamp/interval
param can be one of:
1. A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'.
2. A date value
3. A timestamp value
4. A year-month interval valueExample:

```
 > SELECT month('2009-07-30') FROM src LIMIT 1;
 7
```

Function class:org.apache.hadoop.hive.ql.udf.UDFMonth
Function type:BUILTIN

months_between

months_between(date1, date2, roundOff) - returns number of months between dates date1 and date2
If date1 is later than date2, then the result is positive. If date1 is earlier than date2, then the result is negative. If date1 and date2 are either the same days of the month or both last days of months, then the result is always an integer. Otherwise the UDF calculates the fractional portion of the result based on a 31-day month and considers the difference in time components date1 and date2.
date1 and date2 type can be date, timestamp or string in the format 'yyyy-MM-dd' or 'yyyy-MM-dd HH:mm:ss'. The result is rounded to 8 decimal places by default. Set roundOff=false otherwise.
Example:

```
 > SELECT months_between('1997-02-28 10:30:00', '1996-10-30');
3.94959677
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMonthsBetween
Function type:BUILTIN

murmur_hash

murmur_hash(a1, a2, ...) - Returns a hash value of the arguments
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFMurmurHash
Function type:BUILTIN

named_struct

named_struct(name1, val1, name2, val2, ...) - Creates a struct with the given field names and values
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFNamedStruct
Function type:BUILTIN

negative

negative a - Returns -a
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNegative
Function type:BUILTIN

next_day

next_day(start_date, day_of_week) - Returns the first date which is later than start_date and named as indicated.
start_date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. day_of_week is day of the week (e.g. Mo, tue, FRIDAY).Example:

```
 > SELECT next_day('2015-01-14', 'TU') FROM src LIMIT 1;
'2015-01-20'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFNextDay
Function type:BUILTIN

ngrams

ngrams(expr, n, k, pf) - Estimates the top-k n-grams in rows that consist of sequences of strings, represented as arrays of strings, or arrays of arrays of strings. 'pf' is an optional precision factor that controls memory usage.
The parameter 'n' specifies what type of n-grams are being estimated. Unigrams are n = 1, and bigrams are n = 2. Generally, n will not be greater than about 5. The 'k' parameter specifies how many of the highest-frequency n-grams will be returned by the UDAF. The optional precision factor 'pf' specifies how much memory to use for estimation; more memory will give more accurate frequency counts, but could crash the JVM. The default value is 20, which

internally maintains 20*k n-grams, but only returns the k highest frequency ones. The output is an array of structs with the top-k n-grams. It might be convenient to explode() the output of this UDAF.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFnGrams
Function type:BUILTIN

noop

There is no documentation for function 'noop'
Function class:org.apache.hadoop.hive.ql.udf.ptf.Noop$NoopResolver
Function type:BUILTIN

noopstreaming

There is no documentation for function 'noopstreaming'
Function class:org.apache.hadoop.hive.ql.udf.ptf.NoopStreaming$NoopStreamingResolver
Function type:BUILTIN

noopwithmap

There is no documentation for function 'noopwithmap'
Function class:org.apache.hadoop.hive.ql.udf.ptf.NoopWithMap$NoopWithMapResolver
Function type:BUILTIN

noopwithmapstreaming

There is no documentation for function 'noopwithmapstreaming'
Function class:org.apache.hadoop.hive.ql.udf.ptf.NoopWithMapStreaming$NoopWithMapStreamingResolver
Function type:BUILTIN

not

not a - Logical not
Synonyms: !
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPNot
Function type:BUILTIN

ntile

There is no documentation for function 'ntile'
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFNTile
Function type:BUILTIN

nullif

nullif(a1, a2) - shorthand for: case when a1 = a2 then null else a1
Example:

SELECT nullif(1,1),nullif(1,2)

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFNullif
Function type:BUILTIN

nvl

nvl(value,default_value) - Returns default value if value is null else returns value
Example:

> SELECT nvl(null,'bla') FROM src LIMIT 1;
bla

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFNvl
Function type:BUILTIN

octet_length

octet_length(str | binary) - Returns the number of bytes in str or binary data
Example:

> SELECT octet_length('HUX8 □ ') FROM src LIMIT 1;
15

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOctetLength
Function type:BUILTIN

or

a1 or a2 or ... or an - Logical or
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPOr
Function type:BUILTIN

parse_url

parse_url(url, partToExtract[, key]) - extracts a part from a URL
Parts: HOST, PATH, QUERY, REF, PROTOCOL, AUTHORITY, FILE, USERINFO
key specifies which query to extract
Example:

```
 > SELECT parse_url('http://facebook.com/path/p1.php?query=1', 'HOST') FROM src LIMIT 1;
 'facebook.com'
 > SELECT parse_url('http://facebook.com/path/p1.php?query=1', 'QUERY') FROM src LIMIT 1;
 'query=1'
 > SELECT parse_url('http://facebook.com/path/p1.php?query=1', 'QUERY', 'query') FROM src LI
MIT 1;
 '1'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFParseUrl
Function type:BUILTIN

parse_url_tuple

parse_url_tuple(url, partname1, partname2, ..., partnameN) - extracts N (N>=1) parts from a URL.
It takes a URL and one or multiple partnames, and returns a tuple. All the input parameters and output column types are string.
Partname: HOST, PATH, QUERY, REF, PROTOCOL, AUTHORITY, FILE, USERINFO, QUERY:<key_name>
Note: Partnames are case-sensitive, and should not contain unnecessary white spaces.
Example:

```
 > SELECT b.* FROM src LATERAL VIEW parse_url_tuple(fullurl, 'HOST', 'PATH', 'QUERY', 'QUERY:
id') b as host, path, query, query_id LIMIT 1;
 > SELECT parse_url_tuple(a.fullurl, 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'FILE',  'AUTHO
RITY', 'USERINFO', 'QUERY:k1') as (ho, pa, qu, re, pr, fi, au, us, qk1) from src a;
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFParseUrlTuple
Function type:BUILTIN

percent_rank

There is no documentation for function 'percent_rank'
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFPercentRank
Function type:BUILTIN

percentile

percentile(expr, pc) - Returns the percentile(s) of expr at pc (range: [0,1]).pc can be a double or double array
Function class:org.apache.hadoop.hive.ql.udf.UDAFPercentile
Function type:BUILTIN

percentile_approx

percentile_approx(expr, pc, [nb]) - For very large data, computes an approximate percentile value from a histogram, using the optional argument [nb] as the number of histogram bins to use. A higher value of nb results in a more accurate approximation, at the cost of higher memory usage.
'expr' can be any numeric column, including doubles and floats, and 'pc' is either a single double/float with a requested percentile, or an array of double/float with multiple percentiles. If 'nb' is not specified, the default approximation is done with 10,000 histogram bins, which means that if there are 10,000 or fewer unique values in 'expr', you can expect an exact result. The percentile() function always computes an exact percentile and can run out of memory if there are too many unique values in a column, which necessitates this function.
Example (three percentiles requested using a finer histogram approximation):

> SELECT percentile_approx(val, array(0.5, 0.95, 0.98), 100000) FROM somedata;
[0.05,1.64,2.26]

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFPercentileApprox
Function type:BUILTIN

pi

pi() - returns pi
Example:

> SELECT pi() FROM src LIMIT 1;
3.14159...

Function class:org.apache.hadoop.hive.ql.udf.UDFPI
Function type:BUILTIN

pmod

a pmod b - Compute the positive modulo
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFPosMod
Function type:BUILTIN

posexplode

posexplode(a) - behaves like explode for arrays, but includes the position of items in the original array
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFPosExplode
Function type:BUILTIN

positive

positive a - Returns a
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFOPPositive
Function type:BUILTIN

pow

pow(x1, x2) - raise x1 to the power of x2
Synonyms: power
Example:

```
 > SELECT pow(2, 3) FROM src LIMIT 1;
 8
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFPower
Function type:BUILTIN

power

power(x1, x2) - raise x1 to the power of x2
Synonyms: pow
Example:

```
 > SELECT power(2, 3) FROM src LIMIT 1;
 8
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFPower
Function type:BUILTIN

printf

printf(String format, Obj... args) - function that can format strings according to printf-style format strings
Example:

```
 > SELECT printf("Hello World %d %s", 100, "days")FROM src LIMIT 1;
 "Hello World 100 days"
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFPrintf
Function type:BUILTIN

quarter

quarter(date/timestamp/string) - Returns the quarter of the year for date, in the range 1 to 4.
Example:

```
> SELECT quarter('2015-04-08');
 2
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFQuarter
Function type:BUILTIN

radians

radians(x) - Converts degrees to radians
Example:

```
 > SELECT radians(90) FROM src LIMIT 1;
 1.5707963267949mo
```

Function class:org.apache.hadoop.hive.ql.udf.UDFRadians
Function type:BUILTIN

rand

rand([seed]) - Returns a pseudorandom number between 0 and 1
Function class:org.apache.hadoop.hive.ql.udf.UDFRand
Function type:BUILTIN

rank

There is no documentation for function 'rank'
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFRank
Function type:BUILTIN

reflect

reflect(class,method[,arg1[,arg2..]]) calls method with reflection
Synonyms: java_method
Use this UDF to call Java methods by matching the argument signature

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFReflect
Function type:BUILTIN

reflect2

reflect2(arg0,method[,arg1[,arg2..]]) calls method of arg0 with reflection
Use this UDF to call Java methods by matching the argument signature

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFReflect2
Function type:BUILTIN

regexp

str regexp regexp - Returns true if str matches regexp and false otherwise
Synonyms: rlike
Example:

```
> SELECT 'fb' regexp '.*' FROM src LIMIT 1;
true
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRegExp

Function type:BUILTIN

regexp_extract

regexp_extract(str, regexp[, idx]) - extracts a group that matches regexp
Example:

```
 > SELECT regexp_extract('100-200', '(₩d+)-(₩d+)', 1) FROM src LIMIT 1;
 '100'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFRegExpExtract
Function type:BUILTIN

regexp_replace

regexp_replace(str, regexp, rep) - replace all substrings of str that match regexp with rep
Example:

```
 > SELECT regexp_replace('100-200', '(₩d+)', 'num') FROM src LIMIT 1;
 'num-num'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFRegExpReplace
Function type:BUILTIN

regr_avgx

regr_avgx(y,x) - evaluates the average of the independent variable
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
Otherwise, it computes the following:
AVG(X)
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrAvgX
Function type:BUILTIN

regr_avgy

regr_avgy(y,x) - evaluates the average of the dependent variable

The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
Otherwise, it computes the following:
AVG(Y)
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrAvgY
Function type:BUILTIN

regr_count

regr_count(y,x) - returns the number of non-null pairs
The function takes as arguments any pair of numeric types and returns a long.
Any pair with a NULL is ignored.
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrCount
Function type:BUILTIN

regr_intercept

regr_intercept(y,x) - returns the y-intercept of the regression line.
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
If N*SUM(x*x) = SUM(x)*SUM(x): NULL is returned.
Otherwise, it computes the following:
(SUM(y)*SUM(x*x)-SUM(X)*SUM(x*y)) / (N*SUM(x*x)-SUM(x)*SUM(x))
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrIntercept
Function type:BUILTIN

regr_r2

regr_r2(y,x) - returns the coefficient of determination (also called R-squared or goodness of fit) for the regression line.
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
If N*SUM(x*x) = SUM(x)*SUM(x): NULL is returned.
If N*SUM(y*y) = SUM(y)*SUM(y): 1 is returned.
Otherwise, it computes the following:
POWER(N*SUM(x*y)-SUM(x)*SUM(y) ,2) / ((N*SUM(x*x)-SUM(x)*SUM(x)) *
(N*SUM(y*y)-SUM(y)*SUM(y)))
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrR2
Function type:BUILTIN

regr_slope

regr_slope(y,x) - returns the slope of the linear regression line
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
If N*SUM(x*x) = SUM(x)*SUM(x): NULL is returned (the fit would be a vertical).
Otherwise, it computes the following:
(N*SUM(x*y)-SUM(x)*SUM(y)) / (N*SUM(x*x)-SUM(x)*SUM(x))
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrSlope
Function type:BUILTIN

regr_sxx

regr_sxx(y,x) - auxiliary analytic function
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
Otherwise, it computes the following:
SUM(x*x)-SUM(x)*SUM(x)/N

Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrSXX
Function type:BUILTIN

regr_sxy

regr_sxy(y,x) - return a value that can be used to evaluate the statistical validity of a regression model.
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.
If applied to an empty set: NULL is returned.
If N*SUM(x*x) = SUM(x)*SUM(x): NULL is returned.
Otherwise, it computes the following:
SUM(x*y)-SUM(x)*SUM(y)/N
Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrSXY
Function type:BUILTIN

regr_syy

regr_syy(y,x) - auxiliary analytic function
The function takes as arguments any pair of numeric types and returns a double.
Any pair with a NULL is ignored.

If applied to an empty set: NULL is returned.
Otherwise, it computes the following:
SUM(y*y)-SUM(y)*SUM(y)/N

Function
class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFBinarySetFunctions$RegrSYY
Function type:BUILTIN

repeat

repeat(str, n) - repeat str n times
Example:

```
 > SELECT repeat('123', 2) FROM src LIMIT 1;
 '123123'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFRepeat
Function type:BUILTIN

replace

replace(str, search, rep) - replace all substrings of 'str' that match 'search' with 'rep'
Example:

```
 > SELECT replace('Hack and Hue', 'H', 'BL') FROM src LIMIT 1;
 'BLack and BLue'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFReplace
Function type:BUILTIN

replicate_rows

replicate_rows(n, cols...) - turns 1 row into n rows
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFReplicateRows
Function type:BUILTIN

restrict_information_schema

restrict_information_schema() - Returns whether or not to enable information schema

restriction. Currently it is enabled if either HS2 authorizer or metastore authorizer implements policy provider interface.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRestrictInformationSchema
Function type:BUILTIN

reverse

reverse(str) - reverse str
Example:

```
 > SELECT reverse('Facebook') FROM src LIMIT 1;
 'koobecaF'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFReverse
Function type:BUILTIN

rlike

str rlike regexp - Returns true if str matches regexp and false otherwise
Synonyms: regexp
Example:

```
 > SELECT 'fb' rlike '.*' FROM src LIMIT 1;
 true
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRegExp
Function type:BUILTIN

round

round(x[, d]) - round x to d decimal places
Example:

```
 > SELECT round(12.3456, 1) FROM src LIMIT 1;
 12.3'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRound
Function type:BUILTIN

row_number

row_number - The ROW_NUMBER function assigns a unique number (sequentially, starting from 1, as defined by ORDER BY) to each row within the partition.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFRowNumber
Function type:BUILTIN

rpad

rpad(str, len, pad) - Returns str, right-padded with pad to a length of len
If str is longer than len, the return value is shortened to len characters.
In case of empty pad string, the return value is null.
Example:

```
 > SELECT rpad('hi', 5, '??') FROM src LIMIT 1;
 'hi???'
 > SELECT rpad('hi', 1, '??') FROM src LIMIT 1;
 'h'
 > SELECT rpad('hi', 5, '') FROM src LIMIT 1;
 null
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRpad
Function type:BUILTIN

rtrim

rtrim(str) - Removes the trailing space characters from str
Example:

```
 > SELECT rtrim('facebook   ') FROM src LIMIT 1;
 'facebook'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFRTrim
Function type:BUILTIN

second

second(date) - Returns the second component of the string/timestamp/interval
param can be one of:
1. A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'HH:mm:ss'.
2. A timestamp value
3. A day-time interval value.
Example:

```
 > SELECT second('2009-07-30 12:58:59') FROM src LIMIT 1;
59
> SELECT second('12:58:59') FROM src LIMIT 1;
59
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSecond
Function type:BUILTIN

sentences

sentences(str, lang, country) - Splits str into arrays of sentences, where each sentence is an array of words. The 'lang' and'country' arguments are optional, and if omitted, the default locale is used.
Example:

```
 > SELECT sentences('Hello there! I am a UDF.') FROM src LIMIT 1;
[ ["Hello", "there"], ["I", "am", "a", "UDF"] ]
 > SELECT sentences(review, language) FROM movies;
```

Unnecessary punctuation, such as periods and commas in English, is automatically stripped. If specified, 'lang' should be a two-letter ISO-639 language code (such as 'en'), and 'country' should be a two-letter ISO-3166 code (such as 'us'). Not all country and language codes are fully supported, and if an unsupported code is specified, a default locale is used to process that string.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSentences
Function type:BUILTIN

sha

sha(str or bin) - Calculates the SHA-1 digest for string or binary and returns the value as a hex string.
Synonyms: sha1

Example:

```
  > SELECT sha('ABC');
  '3c01bdbb26f358bab27f267924aa2c9a03fcfdb8'
  > SELECT sha(binary('ABC'));
  '3c01bdbb26f358bab27f267924aa2c9a03fcfdb8'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSha1
Function type:BUILTIN

sha1

sha1(str or bin) - Calculates the SHA-1 digest for string or binary and returns the value as a hex string.
Synonyms: sha
Example:

```
  > SELECT sha1('ABC');
  '3c01bdbb26f358bab27f267924aa2c9a03fcfdb8'
  > SELECT sha1(binary('ABC'));
  '3c01bdbb26f358bab27f267924aa2c9a03fcfdb8'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSha1
Function type:BUILTIN

sha2

sha2(string/binary, len) - Calculates the SHA-2 family of hash functions (SHA-224, SHA-256, SHA-384, and SHA-512).
The first argument is the string or binary to be hashed. The second argument indicates the desired bit length of the result, which must have a value of 224, 256, 384, 512, or 0 (which is equivalent to 256). SHA-224 is supported starting from Java 8. If either argument is NULL or the hash length is not one of the permitted values, the return value is NULL.
Example:

```
> SELECT sha2('ABC', 256);
 'b5d4045c3f466fa91fe2cc6abe79232a1a57cdf104f7a26e716e0a1e2789df78'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSha2
Function type:BUILTIN

shiftleft

shiftleft(a, b) - Bitwise left shift
Returns int for tinyint, smallint and int a. Returns bigint for bigint a.
Example:

```
 > SELECT shiftleft(2, 1);
 4
```

Function class:org.apache.hadoop.hive.ql.udf.UDFOPBitShiftLeft
Function type:BUILTIN

shiftright

shiftright(a, b) - Bitwise right shift
Returns int for tinyint, smallint and int a. Returns bigint for bigint a.
Example:

```
 > SELECT shiftright(4, 1);
 2
```

Function class:org.apache.hadoop.hive.ql.udf.UDFOPBitShiftRight
Function type:BUILTIN

shiftrightunsigned

shiftrightunsigned(a, b) - Bitwise unsigned right shift
Returns int for tinyint, smallint and int a. Returns bigint for bigint a.
Example:

```
 > SELECT shiftrightunsigned(4, 1);
 2
```

Function class:org.apache.hadoop.hive.ql.udf.UDFOPBitShiftRightUnsigned
Function type:BUILTIN

sign

sign(x) - returns the sign of x)
Example:

```
  > SELECT sign(40) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSign
Function type:BUILTIN

sin

sin(x) - returns the sine of x (x is in radians)
Example:

```
  > SELECT sin(0) FROM src LIMIT 1;
 0
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSin
Function type:BUILTIN

size

size(a) - Returns the size of a
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSize
Function type:BUILTIN

sort_array

sort_array(array(obj1, obj2,...)) - Sorts the input array in ascending order according to the natural ordering of the array elements.
Example:

```
  > SELECT sort_array(array('b', 'd', 'c', 'a')) FROM src LIMIT 1;
  'a', 'b', 'c', 'd'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSortArray
Function type:BUILTIN

sort_array_by

sort_array_by(array(obj1, obj2,...),'f1','f2',...,['ASC','DESC']) - Sorts the input tuple array in user specified order(ASC,DESC) by desired field[s] name If sorting order is not mentioned by user then dafault sorting order is ascending
Example:

```
 > SELECT sort_array_by(array(struct('g',100),struct('b',200)),'col1','ASC') FROM src LIMIT 1;
array(struct('b',200),struct('g',100))
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSortArrayByField
Function type:BUILTIN

soundex

soundex(string) - Returns soundex code of the string.
The soundex code consist of the first letter of the name followed by three digits.
Example:

```
> SELECT soundex('Miller');
M460
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSoundex
Function type:BUILTIN

space

space(n) - returns n spaces
Example:

```
  > SELECT space(2) FROM src LIMIT 1;
' '
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSpace
Function type:BUILTIN

split

split(str, regex) - Splits str around occurances that match regex
Example:

```
 > SELECT split('oneAtwoBthreeC', '[ABC]') FROM src LIMIT 1;
 ["one", "two", "three"]
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSplit
Function type:BUILTIN

sq_count_check

sq_count_check(x) - Internal check on scalar subquery expression to make sure atmost one row is returned
For internal use only
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSQCountCheck
Function type:BUILTIN

sqrt

sqrt(x) - returns the square root of x
Example:

```
  > SELECT sqrt(4) FROM src LIMIT 1;
 2
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSqrt
Function type:BUILTIN

stack

stack(n, cols...) - turns k columns into n rows of size k/n each
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDTFStack
Function type:BUILTIN

std

std(x) - Returns the standard deviation of a set of numbers
Synonyms: stddev, stddev_pop
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFStd
Function type:BUILTIN

stddev

stddev(x) - Returns the standard deviation of a set of numbers
Synonyms: std, stddev_pop
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFStd
Function type:BUILTIN

stddev_pop

stddev_pop(x) - Returns the standard deviation of a set of numbers
Synonyms: std, stddev
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFStd
Function type:BUILTIN

stddev_samp

stddev_samp(x) - Returns the sample standard deviation of a set of numbers.
If applied to an empty set: NULL is returned.
If applied to a set with a single element: NULL is returned.
Otherwise it computes: sqrt(var_samp(x))
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFStdSample
Function type:BUILTIN

str_to_map

str_to_map(text, delimiter1, delimiter2) - Creates a map by parsing text
Split text into key-value pairs using two delimiters. The first delimiter separates pairs, and the second delimiter sperates key and value. If only one parameter is given, default delimiters are used: ',' as delimiter1 and ':' as delimiter2.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFStringToMap
Function type:BUILTIN

struct

struct(col1, col2, col3, ...) - Creates a struct with the given field values
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFStruct
Function type:BUILTIN

substr

substr(str, pos[, len]) - returns the substring of str that starts at pos and is of length len orsubstr(bin, pos[, len]) - returns the slice of byte array that starts at pos and is of length len
Synonyms: substring
pos is a 1-based index. If pos<0 the starting position is determined by counting backwards from the end of str.
Example:

```
 > SELECT substr('Facebook', 5) FROM src LIMIT 1;
'book'
> SELECT substr('Facebook', -5) FROM src LIMIT 1;
'ebook'
> SELECT substr('Facebook', 5, 1) FROM src LIMIT 1;
'b'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSubstr
Function type:BUILTIN

substring

substring(str, pos[, len]) - returns the substring of str that starts at pos and is of length len orsubstring(bin, pos[, len]) - returns the slice of byte array that starts at pos and is of length len
Synonyms: substr
pos is a 1-based index. If pos<0 the starting position is determined by counting backwards from the end of str.
Example:

```
 > SELECT substring('Facebook', 5) FROM src LIMIT 1;
'book'
> SELECT substring('Facebook', -5) FROM src LIMIT 1;
'ebook'
> SELECT substring('Facebook', 5, 1) FROM src LIMIT 1;
'b'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFSubstr
Function type:BUILTIN

substring_index

substring_index(str, delim, count) - Returns the substring from string str before count occurrences of the delimiter delim.
If count is positive, everything to the left of the final delimiter (counting from the left) is returned. If count is negative, everything to the right of the final delimiter (counting from the right) is returned. Substring_index performs a case-sensitive match when searching for delim.
Example:

```
 > SELECT substring_index('www.apache.org', '.', 2);
 'www.apache'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFSubstringIndex
Function type:BUILTIN

sum

sum(x) - Returns the sum of a set of numbers
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFSum
Function type:BUILTIN

tan

tan(x) - returns the tangent of x (x is in radians)
Example:

```
  > SELECT tan(0) FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFTan
Function type:BUILTIN

to_date

to_date(expr) - Extracts the date part of the date or datetime expression expr
Example:

```
  > SELECT to_date('2009-07-30 04:17:52') FROM src LIMIT 1;
'2009-07-30'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFDate
Function type:BUILTIN

to_epoch_milli

There is no documentation for function 'to_epoch_milli'
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFEpochMilli
Function type:BUILTIN

to_unix_timestamp

to_unix_timestamp(date[, pattern]) - Returns the UNIX timestamp
Converts the specified time to number of seconds since 1970-01-01.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFToUnixTimeStamp
Function type:BUILTIN

to_utc_timestamp

to_utc_timestamp(timestamp, string timezone) - Assumes given timestamp is in given timezone and converts to UTC (as of Hive 0.8.0)
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFToUtcTimestamp
Function type:BUILTIN

translate

translate(input, from, to) - translates the input string by replacing the characters present in the from string with the corresponding characters in the to string
translate(string input, string from, string to) is an equivalent function to translate in PostGreSQL. It works on a character by character basis on the input string (first parameter). A character in the input is checked for presence in the from string (second parameter). If a match happens, the character from to string (third parameter) which appears at the same index as the character in from string is obtained. This character is emitted in the output string instead of the original character from the input string. If the to string is shorter than the from string, there may not be a character present at the same index in the to string. In such a case, nothing is emitted for the original character and it's deleted from the output string.
For example,

translate('abcdef', 'adc', '19') returns '1b9ef' replacing 'a' with '1', 'd' with '9' and removing 'c' from the input string

translate('a b c d', ' ', '') return 'abcd' removing all spaces from the input string

If the same character is present multiple times in the input string, the first occurence of the character is the one that's considered for matching. However, it is not recommended to have the same character more than once in the from string since it's not required and adds to confusion.

For example,

translate('abcdef', 'ada', '192') returns '1bc9ef' replaces 'a' with '1' and 'd' with '9' ignoring the second occurence of 'a' in the from string mapping it to '2'
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFTranslate
Function type:BUILTIN

trim

trim(str) - Removes the leading and trailing space characters from str
Example:

```
 > SELECT trim('   facebook  ') FROM src LIMIT 1;
 'facebook'
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFTrim
Function type:BUILTIN

trunc

trunc(date, fmt) / trunc(N,D) - Returns If input is date returns date with the time portion of the day truncated to the unit specified by the format model fmt. If you omit fmt, then date is truncated to the nearest day. It currently only supports 'MONTH'/'MON'/'MM', 'QUARTER'/'Q' and 'YEAR'/'YYYY'/'YY' as format.If input is a number group returns N truncated to D decimal places. If D is omitted, then N is truncated to 0 places.D can be negative to truncate (make zero) D digits left of the decimal point.
date is a string in the format 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'. The time part of date is ignored.
Example:

```
 > SELECT trunc('2009-02-12', 'MM');
OK
 '2009-02-01'
 > SELECT trunc('2017-03-15', 'Q');
OK
```

'2017-01-01'
> SELECT trunc('2015-10-27', 'YEAR');
OK
'2015-01-01' > SELECT trunc(1234567891.1234567891,4);
OK
1234567891.1234
> SELECT trunc(1234567891.1234567891,-4);
OK
1234560000 > SELECT trunc(1234567891.1234567891,0);
OK
1234567891
> SELECT trunc(1234567891.1234567891);
OK
1234567891

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFTrunc
Function type:BUILTIN

ucase

ucase(str) - Returns str with all characters changed to uppercase
Synonyms: upper
Example:

> SELECT ucase('Facebook') FROM src LIMIT 1;
'FACEBOOK'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFUpper
Function type:BUILTIN

udftoboolean

There is no documentation for function 'udftoboolean'
Function class:org.apache.hadoop.hive.ql.udf.UDFToBoolean
Function type:BUILTIN

udftobyte

There is no documentation for function 'udftobyte'

Function class:org.apache.hadoop.hive.ql.udf.UDFToByte
Function type:BUILTIN

udftodouble

There is no documentation for function 'udftodouble'
Function class:org.apache.hadoop.hive.ql.udf.UDFToDouble
Function type:BUILTIN

udftofloat

There is no documentation for function 'udftofloat'
Function class:org.apache.hadoop.hive.ql.udf.UDFToFloat
Function type:BUILTIN

udftointeger

There is no documentation for function 'udftointeger'
Function class:org.apache.hadoop.hive.ql.udf.UDFToInteger
Function type:BUILTIN

udftolong

There is no documentation for function 'udftolong'
Function class:org.apache.hadoop.hive.ql.udf.UDFToLong
Function type:BUILTIN

udftoshort

There is no documentation for function 'udftoshort'
Function class:org.apache.hadoop.hive.ql.udf.UDFToShort
Function type:BUILTIN

udftostring

There is no documentation for function 'udftostring'
Function class:org.apache.hadoop.hive.ql.udf.UDFToString
Function type:BUILTIN

unbase64

unbase64(str) - Convert the argument from a base 64 string to binary
Function class:org.apache.hadoop.hive.ql.udf.UDFUnbase64
Function type:BUILTIN

unhex

unhex(str) - Converts hexadecimal argument to binary
Performs the inverse operation of HEX(str). That is, it interprets
each pair of hexadecimal digits in the argument as a number and
converts it to the byte representation of the number. The
resulting characters are returned as a binary string.

Example:

> SELECT DECODE(UNHEX('4D7953514C'), 'UTF-8') from src limit 1;
'MySQL'

The characters in the argument string must be legal hexadecimal
digits: '0' .. '9', 'A' .. 'F', 'a' .. 'f'. If UNHEX() encounters
any nonhexadecimal digits in the argument, it returns NULL. Also,
if there are an odd number of characters a leading 0 is appended.
Function class:org.apache.hadoop.hive.ql.udf.UDFUnhex
Function type:BUILTIN

unix_timestamp

unix_timestamp(date[, pattern]) - Converts the time to a number
Converts the specified time to number of seconds since 1970-01-01. The unix_timestamp(void)
overload is deprecated, use current_timestamp.
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFUnixTimeStamp
Function type:BUILTIN

upper

upper(str) - Returns str with all characters changed to uppercase
Synonyms: ucase
Example:

> SELECT upper('Facebook') FROM src LIMIT 1;
'FACEBOOK'

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFUpper
Function type:BUILTIN

uuid

uuid() - Returns a universally unique identifier (UUID) string.
The value is returned as a canonical UUID 36-character string.
Example:

```
 > SELECT uuid();
 '0baf1f52-53df-487f-8292-99a03716b688'
 > SELECT uuid();
 '36718a53-84f5-45d6-8796-4f79983ad49d'
```

Function class:org.apache.hadoop.hive.ql.udf.UDFUUID
Function type:BUILTIN

var_pop

var_pop(x) - Returns the variance of a set of numbers
Synonyms: variance
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFVariance
Function type:BUILTIN

var_samp

var_samp(x) - Returns the sample variance of a set of numbers.
If applied to an empty set: NULL is returned.
If applied to a set with a single element: NULL is returned.
Otherwise it computes: (S2-S1*S1/N)/(N-1)
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFVarianceSample
Function type:BUILTIN

variance

variance(x) - Returns the variance of a set of numbers
Synonyms: var_pop
Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDAFVariance
Function type:BUILTIN

version

version() - Returns the Hive build version string - includes base version and revision.
Function class:org.apache.hadoop.hive.ql.udf.UDFVersion
Function type:BUILTIN

weekofyear

weekofyear(date) - Returns the week of the year of the given date. A week is considered to start on a Monday and week 1 is the first week with >3 days.
Examples:

```
 > SELECT weekofyear('2008-02-20') FROM src LIMIT 1;
 8
 > SELECT weekofyear('1980-12-31 12:59:59') FROM src LIMIT 1;
 1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFWeekOfYear
Function type:BUILTIN

when

CASE WHEN a THEN b [WHEN c THEN d]* [ELSE e] END - When a = true, returns b; when c = true, return d; else return e
Example:

```
SELECT
CASE
  WHEN deptno=1 THEN Engineering
  WHEN deptno=2 THEN Finance
  ELSE admin
END,
CASE
  WHEN zone=7 THEN Americas
  ELSE Asia-Pac
END
FROM emp_details
```

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFWhen
Function type:BUILTIN

width_bucket

width_bucket(expr, min_value, max_value, num_buckets) - Returns an integer between 0 and num_buckets+1 by mapping the expr into buckets defined by the range [min_value,

max_value]
Returns an integer between 0 and num_buckets+1 by mapping expr into the ith equally sized bucket. Buckets are made by dividing [min_value, max_value] into equally sized regions. If expr max_value return num_buckets+1
Example: expr is an integer column withs values 1, 10, 20, 30.

 > SELECT width_bucket(expr, 5, 25, 4) FROM src;
1
1
3
5

Function class:org.apache.hadoop.hive.ql.udf.generic.GenericUDFWidthBucket
Function type:BUILTIN

windowingtablefunction

There is no documentation for function 'windowingtablefunction'
Function class:org.apache.hadoop.hive.ql.udf.ptf.WindowingTableFunction$WindowingTableFunctionResolver
Function type:BUILTIN

xpath

xpath(xml, xpath) - Returns a string array of values within xml nodes that match the xpath expression
Example:

 > SELECT xpath('<a>b1b2b3<c>c1</c><c>c2</c>', 'a/text()') FROM src LIMIT 1
 []
 > SELECT xpath('<a>b1b2b3<c>c1</c><c>c2</c>', 'a/b/text()') FROM src LIMIT 1
 ["b1","b2","b3"]
 > SELECT xpath('<a>b1b2b3<c>c1</c><c>c2</c>', 'a/c/text()') FROM src LIMIT 1
 ["c1","c2"]

Function class:org.apache.hadoop.hive.ql.udf.xml.GenericUDFXPath

Function type:BUILTIN

xpath_boolean

xpath_boolean(xml, xpath) - Evaluates a boolean xpath expression
Example:

```
> SELECT xpath_boolean('<a><b>1</b></a>','a/b') FROM src LIMIT 1;
true
> SELECT xpath_boolean('<a><b>1</b></a>','a/b = 2') FROM src LIMIT 1;
false
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathBoolean
Function type:BUILTIN

xpath_double

xpath_double(xml, xpath) - Returns a double value that matches the xpath expression
Synonyms: xpath_number
Example:

```
> SELECT xpath_double('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
3.0
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathDouble
Function type:BUILTIN

xpath_float

xpath_float(xml, xpath) - Returns a float value that matches the xpath expression
Example:

```
> SELECT xpath_float('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
3.0
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathFloat

Function type:BUILTIN

xpath_int

xpath_int(xml, xpath) - Returns an integer value that matches the xpath expression
Example:

```
 > SELECT xpath_int('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
 3
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathInteger
Function type:BUILTIN

xpath_long

xpath_long(xml, xpath) - Returns a long value that matches the xpath expression
Example:

```
 > SELECT xpath_long('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
 3
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathLong
Function type:BUILTIN

xpath_number

xpath_number(xml, xpath) - Returns a double value that matches the xpath expression
Synonyms: xpath_double
Example:

```
 > SELECT xpath_number('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
 3.0
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathDouble
Function type:BUILTIN

xpath_short

xpath_short(xml, xpath) - Returns a short value that matches the xpath expression
Example:

```
 > SELECT xpath_short('<a><b>1</b><b>2</b></a>','sum(a/b)') FROM src LIMIT 1;
 3
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathShort
Function type:BUILTIN

xpath_string

xpath_string(xml, xpath) - Returns the text contents of the first xml node that matches the xpath expression
Example:

```
 > SELECT xpath_string('<a><b>b</b><c>cc</c></a>','a/c') FROM src LIMIT 1;
 'cc'
 > SELECT xpath_string('<a><b>b1</b><b>b2</b></a>','a/b') FROM src LIMIT 1;
 'b1'
 > SELECT xpath_string('<a><b>b1</b><b>b2</b></a>','a/b[2]') FROM src LIMIT 1;
 'b2'
 > SELECT xpath_string('<a><b>b1</b><b>b2</b></a>','a') FROM src LIMIT 1;
 'b1b2'
```

Function class:org.apache.hadoop.hive.ql.udf.xml.UDFXPathString
Function type:BUILTIN

year

year(param) - Returns the year component of the date/timestamp/interval
param can be one of:
1. A string in the format of 'yyyy-MM-dd HH:mm:ss' or 'yyyy-MM-dd'.
2. A date value
3. A timestamp value
4. A year-month interval value.
Example:

```
 > SELECT year('2009-07-30') FROM src LIMIT 1;
2009
```

Function class:org.apache.hadoop.hive.ql.udf.UDFYear
Function type:BUILTIN

|

a | b - Bitwise or
Example:

```
 > SELECT 3 | 5 FROM src LIMIT 1;
 7
```

Function class:org.apache.hadoop.hive.ql.udf.UDFOPBitOr
Function type:BUILTIN

~

~ n - Bitwise not
Example:

```
 > SELECT ~ 0 FROM src LIMIT 1;
 -1
```

Function class:org.apache.hadoop.hive.ql.udf.UDFOPBitNot
Function type:BUILTIN