

hive常用函数

2017-10-27

hive

关系运算
数学运算
逻辑运算
数值计算
日期函数
条件函数
字符串函数
集合统计函数
复合类型构建操作
复杂类型访问操作
复杂类型长度统计函数

关系运算

1、等值比较: =

语法：A=B

操作类型：所有基本类型

描述: 如果表达式A与表达式B相等，则为TRUE；否则为FALSE

```
hive> select 1 from mytable where 1=1;  
1
```

2、不等值比较: <>

语法: A <> B

操作类型: 所有基本类型

描述: 如果表达式A为NULL，或者表达式B为NULL，返回NULL；如果表达式A与表达式B不相等，则为TRUE；否则为FALSE

```
hive> select 1 from mytable where 1 <> 2;  
1
```

3、小于比较: <

语法: A < B

操作类型: 所有基本类型

描述: 如果表达式A为NULL, 或者表达式B为NULL, 返回NULL; 如果表达式A小于表达式B, 则为TRUE; 否则为FALSE

```
hive> select 1 from mytable where 1 < 2;  
1
```

4、小于等于比较: <=

语法: A <= B

操作类型: 所有基本类型

描述: 如果表达式A为NULL, 或者表达式B为NULL, 返回NULL; 如果表达式A小于或者等于表达式B, 则为TRUE; 否则为FALSE

```
hive> select 1 from mytable where 1 <= 1;  
1
```

5、大于比较: >

语法: A > B

操作类型: 所有基本类型

描述: 如果表达式A为NULL, 或者表达式B为NULL, 返回NULL; 如果表达式A大于表达式B, 则为TRUE; 否则为FALSE

```
hive> select 1 from mytable where 2 > 1;  
1
```

6、大于等于比较: >=

语法: A >= B

操作类型: 所有基本类型

描述: 如果表达式A为NULL, 或者表达式B为NULL, 返回NULL; 如果表达式A大于或者等于表达式B, 则为TRUE; 否则为FALSE

```
hive> select 1 from mytable where 1 >= 1;  
1
```

注意：String的比较要注意(常用的时间比较可以先 to_date 之后再比较)

```
hive> select a, b, a<b, a>b, a=b from mytable;  
2011111209 00:00:00      2011111209      false      true      false
```

7、空值判断: IS NULL

语法: A IS NULL

操作类型: 所有类型

描述: 如果表达式A的值为NULL，则为TRUE；否则为FALSE

```
hive> select 1 from mytable where null is null;  
1
```

8、非空判断: IS NOT NULL

语法: A IS NOT NULL

操作类型: 所有类型

描述: 如果表达式A的值为NULL，则为FALSE；否则为TRUE

```
hive> select 1 from mytable where 1 is not null;  
1
```

9、LIKE比较: LIKE

语法: A LIKE B

操作类型: strings

描述: 如果字符串A或者字符串B为NULL，则返回NULL；如果字符串A符合表达式B 的正则语法，则为TRUE；否则为FALSE。B中字符“_”表示任意单个字符，而字符“%”表示任意数量的字符。

```
hive> select 1 from mytable where 'football' like 'foot%';  
1  
hive> select 1 from mytable where 'football' like 'foot_____';  
1
```

注意：否定比较时候用NOT A LIKE B

```
hive> select 1 from mytable where NOT 'football' like 'fff%';  
1
```

10、JAVA的LIKE操作: RLIKE

语法: A RLIKE B

操作类型: strings

描述: 如果字符串A或者字符串B为NULL，则返回NULL；如果字符串A符合JAVA正则表达式B的正则语法，则为TRUE；否则为FALSE。

```
hive> select 1 from mytable where 'footbar' rlike '^f.*r$';  
1
```

注意：判断一个字符串是否全为数字：

```
hive>select 1 from mytable where '123456' rlike '^\\d+$';  
1  
hive> select 1 from mytable where '123456aa' rlike '^\\d+$';
```

11、REGEXP操作: REGEXP

语法: A REGEXP B

操作类型: strings

描述: 功能与RLIKE相同

```
hive> select 1 from mytable where 'footbar' REGEXP '^f.*r$';  
1
```

数学运算

1、加法操作: +

语法: A + B

操作类型：所有数值类型

说明：返回A与B相加的结果。结果的数值类型等于A的类型和B的类型的最小父类型。比如，int + int 一般结果为int类型，而 int + double 一般结果为double类型

```
hive> select 1 + 9 from mytable;
10
hive> create table mytable as select 1 + 1.2 from mytable;
hive> describe mytable;
_c0      double
```

2、减法操作: -

语法: $A - B$

操作类型: 所有数值类型

说明: 返回A与B相减的结果。结果的数值类型等于A的类型和B的类型的最小父类型。比如, $\text{int} - \text{int}$ 一般结果为int类型, 而 $\text{int} - \text{double}$ 一般结果为double类型

```
hive> select 10 - 5 from mytable;
5
hive> create table mytable1 as select 5.6 - 4 from mytable;
hive> describe mytable;
_c0      double
```

3、乘法操作: *

语法: $A * B$

操作类型: 所有数值类型

说明: 返回A与B相乘的结果。结果的数值类型等于A的类型和B的类型的最小父类型。注意, 如果A乘以B的结果超过默认结果类型的数值范围, 则需要通过cast将结果转换成范围更大的数值类型

```
hive> select 40 * 5 from mytable;
200
```

4、除法操作: /

语法: A / B

操作类型: 所有数值类型

说明: 返回A除以B的结果。结果的数值类型为double

```
hive> select 40 / 5 from mytable;
8.0
```

注意：hive中最高精度的数据类型是double,只精确到小数点后16位，在做除法运算的时候要特别注意

```
hive>select ceil(28.0/6.999999999999999999) from mytable limit 1;
结果为4
hive>select ceil(28.0/6.999999999999999999) from mytable limit 1;
结果为5
```

5、取余操作: %

语法: A % B

操作类型：所有数值类型

说明：返回A除以B的余数。结果的数值类型等于A的类型和B的类型的最小父类型。

```
hive> select 41 % 5 from mytable;
1
hive> select 8.4 % 4 from mytable;
0.400000000000000036
```

注意：精度在hive中是个很大的问题，类似这样的操作最好通过round指定精度

```
hive> select round(8.4 % 4 , 2) from mytable;
0.4
```

6、位与操作: &

语法: A & B

操作类型：所有数值类型

说明：返回A和B按位进行与操作的结果。结果的数值类型等于A的类型和B的类型的最小父类型。

```
hive> select 4 & 8 from mytable;
0
hive> select 6 & 4 from mytable;
4
```

7、位或操作: |

语法: A | B

操作类型：所有数值类型

说明：返回A和B按位进行或操作的结果。结果的数值类型等于A的类型 and B的类型的最小父类型。

```
hive> select 4 | 8 from mytable;
12
hive> select 6 | 8 from mytable;
14
```

8、位异或操作: ^

语法: A ^ B

操作类型：所有数值类型

说明：返回A和B按位进行异或操作的结果。结果的数值类型等于A的类型 and B的类型的最小父类型。

```
hive> select 4 ^ 8 from mytable;
12
hive> select 6 ^ 4 from mytable;
2
```

9 . 位取反操作: ~

语法: ~A

操作类型：所有数值类型

说明：返回A按位取反操作的结果。结果的数值类型等于A的类型。

```
hive> select ~6 from mytable;
-7
hive> select ~4 from mytable;
-5
```

逻辑运算

1、逻辑与操作: AND

语法: A AND B

操作类型：boolean

说明：如果A和B均为TRUE，则为TRUE；否则为FALSE。如果A为NULL或B为NULL，则为NULL

```
hive> select 1 from mytable where 1=1 and 2=2;  
1
```

2、逻辑或操作: OR

语法: A OR B

操作类型：boolean

说明：如果A为TRUE，或者B为TRUE，或者A和B均为TRUE，则为TRUE；否则为FALSE

```
hive> select 1 from mytable where 1=2 or 2=2;  
1
```

3、逻辑非操作: NOT

语法: NOT A

操作类型：boolean

说明：如果A为FALSE，或者A为NULL，则为TRUE；否则为FALSE

```
hive> select 1 from mytable where not 1=2;  
1
```

数值计算

1、取整函数: round

语法: round(double a)

返回值: BIGINT

说明: 返回double类型的整数值部分（遵循四舍五入）

```
hive> select round(3.1415926) from mytable;  
3  
hive> select round(3.5) from mytable;  
4  
hive> create table mytable as select round(9542.158) from mytable;  
hive> describe mytable;  
_c0      bigint
```


2、指定精度取整函数: round

语法: round(double a, int d)

返回值: DOUBLE

说明: 返回指定精度d的double类型

```
hive> select round(3.1415926,4) from mytable;  
3.1416
```

3、向下取整函数: floor

语法: floor(double a)

返回值: BIGINT

说明: 返回等于或者小于该double变量的最大的整数

```
hive> select floor(3.1415926) from mytable;  
3  
hive> select floor(25) from mytable;  
25
```

4、向上取整函数: ceil

语法: ceil(double a)

返回值: BIGINT

说明: 返回等于或者大于该double变量的最小的整数

```
hive> select ceil(3.1415926) from mytable;  
4  
hive> select ceil(46) from mytable;  
46
```

5、向上取整函数: ceiling

语法: ceiling(double a)

返回值: BIGINT

说明: 与ceil功能相同

```
hive> select ceiling(3.1415926) from mytable;  
4  
hive> select ceiling(46) from mytable;  
46
```

6、取随机数函数: rand

语法: rand(),rand(int seed)

返回值: double

说明: 返回一个0到1范围内的随机数。如果指定种子seed, 则会等到一个稳定的随机数序列

```
hive> select rand() from mytable;  
0.5577432776034763  
hive> select rand() from mytable;  
0.6638336467363424  
hive> select rand(100) from mytable;  
0.7220096548596434  
hive> select rand(100) from mytable;  
0.7220096548596434
```

7、自然指数函数: exp

语法: exp(double a)

返回值: double

说明: 返回自然对数e的a次方

```
hive> select exp(2) from mytable;  
7.38905609893065
```

自然对数函数: ln

语法: ln(double a)

返回值: double

说明: 返回a的自然对数

```
hive> select ln(7.38905609893065) from mytable;  
2.0
```

8、以10为底对数函数: log10

语法: log10(double a)

返回值: double

说明: 返回以10为底的a的对数

```
hive> select log10(100) from mytable;  
2.0
```

9、以2为底对数函数: log2

语法: log2(double a)

返回值: double

说明: 返回以2为底的a的对数

```
hive> select log2(8) from mytable;  
3.0
```

10、对数函数: log

语法: log(double base, double a)

返回值: double

说明: 返回以base为底的a的对数

```
hive> select log(4,256) from mytable;  
4.0
```

11、幂运算函数: pow

语法: pow(double a, double p)

返回值: double

说明: 返回a的p次幂

```
hive> select pow(2,4) from mytable;  
16.0
```

12、幂运算函数: power

语法: power(double a, double p)

返回值: double

说明: 返回a的p次幂,与pow功能相同

```
hive> select power(2,4) from mytable;  
16.0
```

13、开平方函数: sqrt

语法: sqrt(double a)

返回值: double

说明: 返回a的平方根

```
hive> select sqrt(16) from mytable;  
4.0
```

14、二进制函数: bin

语法: bin(BIGINT a)

返回值: string

说明: 返回a的二进制代码表示

```
hive> select bin(7) from mytable;  
111
```

15、十六进制函数: hex

语法: hex(BIGINT a)

返回值: string

说明: 如果变量是int类型，那么返回a的十六进制表示；如果变量是string类型，则返回该字符串的十六进制表示

```
hive> select hex(17) from mytable;  
11  
hive> select hex('abc') from mytable;  
616263
```

16、反转十六进制函数: unhex

语法: unhex(string a)

返回值: string

说明: 返回该十六进制字符串所代码的字符串

```
hive> select unhex( '616263' ) from mytable;
abc
hive> select unhex( '11' ) from mytable;
_
hive> select unhex(616263) from mytable;
abc
```

17、进制转换函数: conv

语法: conv(BIGINT num, int from_base, int to_base)

返回值: string

说明: 将数值num从from_base进制转化到to_base进制

```
hive> select conv(17,10,16) from mytable;
11
hive> select conv(17,10,2) from mytable;
10001
```

18、绝对值函数: abs

语法: abs(double a) abs(int a)

返回值: double int

说明: 返回数值a的绝对值

```
hive> select abs(-3.9) from mytable;
3.9
hive> select abs(10.9) from mytable;
10.9
```

19、正取余函数: pmod

语法: pmod(int a, int b),pmod(double a, double b)

返回值: int double

说明: 返回正的a除以b的余数

```
hive> select pmod(9,4) from mytable;
1
hive> select pmod(-9,4) from mytable;
3
```

20、正弦函数: sin

语法: sin(double a)

返回值: double

说明: 返回a的正弦值

```
hive> select sin(0.8) from mytable;  
0.7173560908995228
```

21、反正弦函数: asin

语法: asin(double a)

返回值: double

说明: 返回a的反正弦值

```
hive> select asin(0.7173560908995228) from mytable;  
0.8
```

22、余弦函数: cos

语法: cos(double a)

返回值: double

说明: 返回a的余弦值

```
hive> select cos(0.9) from mytable;  
0.6216099682706644
```

23、反余弦函数: acos

语法: acos(double a)

返回值: double

说明: 返回a的反余弦值

```
hive> select acos(0.6216099682706644) from mytable;  
0.9
```

24、positive函数: positive

语法: positive(int a), positive(double a)

返回值: int double

说明: 返回a

```
hive> select positive(-10) from mytable;
-10
hive> select positive(12) from mytable;
12
```

25、negative函数: negative

语法: negative(int a), negative(double a)

返回值: int double

说明: 返回-a

```
hive> select negative(-5) from mytable;
5
hive> select negative(8) from mytable;
-8
```

日期函数

1、UNIX时间戳转日期函数: from_unixtime

语法: from_unixtime(bigint unixtime, string format)

返回值: string

说明: 转化UNIX时间戳 (从1970-01-01 00:00:00 UTC到指定时间的秒数) 到当前时区的时间格式

```
hive> select from_unixtime(1323308943, 'yyyyMMdd') from mytable;
20111208
```

2、获取当前UNIX时间戳函数: unix_timestamp

语法: unix_timestamp()

返回值: bigint

说明: 获得当前时区的UNIX时间戳

```
hive> select unix_timestamp() from mytable;
1323309615
```

3、日期转UNIX时间戳函数: unix_timestamp

语法: unix_timestamp(string date)

返回值: bigint

说明: 转换格式为"yyyy-MM-dd HH:mm:ss"的日期到UNIX时间戳。如果转化失败, 则返回0。

```
hive> select unix_timestamp('2011-12-07 13:01:03') from mytable;  
1323234063
```

4、指定格式日期转UNIX时间戳函数: unix_timestamp

语法: unix_timestamp(string date, string pattern)

返回值: bigint

说明: 转换pattern格式的日期到UNIX时间戳。如果转化失败, 则返回0。

```
hive> select unix_timestamp('20111207 13:01:03', 'yyyyMMdd HH:mm:ss') from mytable;  
1323234063
```

5、日期时间转日期函数: to_date

语法: to_date(string timestamp)

返回值: string

说明: 返回日期时间字段中的日期部分。

```
hive> select to_date('2011-12-08 10:03:01') from mytable;  
2011-12-08
```

6、日期转年函数: year

语法: year(string date)

返回值: int

说明: 返回日期中的年。

```
hive> select year('2011-12-08 10:03:01') from mytable;  
2011  
hive> select year('2012-12-08') from mytable;  
2012
```


7、日期转月函数: month

语法: month (string date)

返回值: int

说明: 返回日期中的月份。

```
hive> select month('2011-12-08 10:03:01') from mytable;
12
hive> select month('2011-08-08') from mytable;
8
```

8、日期转天函数: day

语法: day (string date)

返回值: int

说明: 返回日期中的天。

```
hive> select day('2011-12-08 10:03:01') from mytable;
8
hive> select day('2011-12-24') from mytable;
24
```

9、日期转小时函数: hour

语法: hour (string date)

返回值: int

说明: 返回日期中的小时。

```
hive> select hour('2011-12-08 10:03:01') from mytable;
10
```

10、日期转分钟函数: minute

语法: minute (string date)

返回值: int

说明: 返回日期中的分钟。

```
hive> select minute('2011-12-08 10:03:01') from mytable;
3
```

11、日期转秒函数: second

语法: second (string date)

返回值: int

说明: 返回日期中的秒。

```
hive> select second('2011-12-08 10:03:01') from mytable;  
1
```

12、日期转周函数: weekofyear

语法: weekofyear (string date)

返回值: int

说明: 返回日期在当前的周数。

```
hive> select weekofyear('2011-12-08 10:03:01') from mytable;  
49
```

13、日期比较函数: datediff

语法: datediff(string enddate, string startdate)

返回值: int

说明: 返回结束日期减去开始日期的天数。

```
hive> select datediff('2012-12-08','2012-05-09') from mytable;  
213
```

14、日期增加函数: date_add

语法: date_add(string startdate, int days)

返回值: string

说明: 返回开始日期startdate增加days天后的日期。

```
hive> select date_add('2012-12-08',10) from mytable;  
2012-12-18
```

15、日期减少函数: date_sub

语法: date_sub (string startdate, int days)

返回值: string

说明: 返回开始日期startdate减少days天后的日期。

```
hive> select date_sub('2012-12-08',10) from mytable;  
2012-11-28
```

条件函数

1、If函数: if

语法: if(boolean testCondition, T valueTrue, T valueFalseOrNull)

返回值: T

说明: 当条件testCondition为TRUE时, 返回valueTrue; 否则返回valueFalseOrNull

```
hive> select if(1=2,100,200) from mytable;  
200  
hive> select if(1=1,100,200) from mytable;  
100
```

2、非空查找函数: COALESCE

语法: COALESCE(T v1, T v2, ...)

返回值: T

说明: 返回参数中的第一个非空值; 如果所有值都为NULL, 那么返回NULL

```
hive> select COALESCE(null,'100','50') from mytable;  
100
```

3、条件判断函数: CASE

语法: CASE a WHEN b THEN c [WHEN d THEN e]* [ELSE f] END

返回值: T

说明: 如果a等于b, 那么返回c; 如果a等于d, 那么返回e; 否则返回f

```
hive> Select case 100 when 50 then 'tom' when 100 then 'mary' else 'tim' end from  
mytable;  
mary
```

```
hive> Select case 200 when 50 then 'tom' when 100 then 'mary' else 'tim' end from
mytable;
tim
```

4、条件判断函数：CASE

语法: CASE WHEN a THEN b [WHEN c THEN d]* [ELSE e] END

返回值: T

说明：如果a为TRUE,则返回b；如果c为TRUE，则返回d；否则返回e

```
hive> select case when 1=2 then 'tom' when 2=2 then 'mary' else 'tim' end from
mytable;
mary
hive> select case when 1=1 then 'tom' when 2=2 then 'mary' else 'tim' end from
mytable;
tom
```

字符串函数

1、字符串长度函数：length

语法: length(string A)

返回值: int

说明：返回字符串A的长度

```
hive> select length('abcdefg') from mytable;
7
```

2、字符串反转函数：reverse

语法: reverse(string A)

返回值: string

说明：返回字符串A的反转结果

```
hive> select reverse('abcdefg') from mytable;
gfdecba
```

3、字符串连接函数：concat

语法: concat(string A, string B...)

返回值: string

说明：返回输入字符串连接后的结果，支持任意个输入字符串

```
hive> select concat('abc','def','gh') from mytable;  
abcdefgh
```

4、带分隔符字符串连接函数：concat_ws

语法: concat_ws(string SEP, string A, string B...)

返回值: string

说明：返回输入字符串连接后的结果，SEP表示各个字符串间的分隔符

```
hive> select concat_ws(',', 'abc','def','gh') from mytable;  
abc,def,gh
```

5、字符串截取函数：substr,substring

语法: substr(string A, int start),substring(string A, int start)

返回值: string

说明：返回字符串A从start位置到结尾的字符串

```
hive> select substr('abcde',3) from mytable;  
cde  
hive> select substring('abcde',3) from mytable;  
cde  
hive> select substr('abcde',-1) from mytable;    （和ORACLE相同）  
e
```

6、字符串截取函数：substr,substring

语法: substr(string A, int start, int len),substring(string A, int start, int len)

返回值: string

说明：返回字符串A从start位置开始，长度为len的字符串

```
hive> select substr('abcde',3,2) from mytable;  
cd  
hive> select substring('abcde',3,2) from mytable;  
cd  
hive> select substring('abcde',-2,2) from mytable;  
de
```

7、字符串转大写函数：upper,ucase

语法: upper(string A) ucase(string A)

返回值: string

说明：返回字符串A的大写格式

```
hive> select upper('abSEd') from mytable;
ABSED
hive> select ucase('abSEd') from mytable;
ABSED
```

8、字符串转小写函数：lower,lcase

语法: lower(string A) lcase(string A)

返回值: string

说明：返回字符串A的小写格式

```
hive> select lower('abSEd') from mytable;
absed
hive> select lcase('abSEd') from mytable;
absed
```

9、去空格函数：trim

语法: trim(string A)

返回值: string

说明：去除字符串两边的空格

```
hive> select trim(' abc ') from mytable;
abc
```

10、左边去空格函数：ltrim

语法: ltrim(string A)

返回值: string

说明：去除字符串左边的空格

```
hive> select ltrim(' abc ') from mytable;  
abc
```

11、右边去空格函数： rtrim

语法: rtrim(string A)

返回值: string

说明：去除字符串右边的空格

```
hive> select rtrim(' abc ') from mytable;  
abc
```

12、正则表达式替换函数： regexp_replace

语法: regexp_replace(string A, string B, string C)

返回值: string

说明：将字符串A中的符合java正则表达式B的部分替换为C。注意，在有些情况下要使用转义字符,类似oracle中的regexp_replace函数。

```
hive> select regexp_replace('foobar', 'oo|ar', '') from mytable;  
fb
```

13、正则表达式解析函数： regexp_extract

语法: regexp_extract(string subject, string pattern, int index)

返回值: string

说明：将字符串subject按照pattern正则表达式的规则拆分，返回index指定的字符。

```
hive> select regexp_extract('foothebar', 'foo(.*?) (bar)', 1) from mytable;  
the  
hive> select regexp_extract('foothebar', 'foo(.*?) (bar)', 2) from mytable;  
bar  
hive> select regexp_extract('foothebar', 'foo(.*?) (bar)', 0) from mytable;  
foothebar
```

注意，在有些情况下要使用转义字符，下面的等号要用双竖线转义，这是java正则表达式的规则。

```
select data_field,
  regexp_extract(data_field, '.*?bgStart\\=[^&]+)', 1) as aaa,
  regexp_extract(data_field, '.*?contentLoaded_headStart\\=[^&]+)', 1) as bbb,
  regexp_extract(data_field, '.*?AppLoad2Req\\=[^&]+)', 1) as ccc
from pt_nginx_loginlog_st
where pt = '2012-03-26' limit 2;
```

14、URL解析函数：parse_url

语法: parse_url(string urlString, string partToExtract [, string keyToExtract])

返回值: string

说明：返回URL中指定的部分。partToExtract的有效值为：HOST, PATH, QUERY, REF, PROTOCOL, AUTHORITY, FILE, and USERINFO.

```
hive> select parse_url('https://www.mytable.com/path1/p.php?k1=v1&k2=v2#Ref1',
'HOST') from mytable;
facebook.com
hive> select parse_url('https://www.mytable.com/path1/p.php?k1=v1&k2=v2#Ref1',
'QUERY', 'k1') from mytable;
v1
```

15、json解析函数：get_json_object

语法: get_json_object(string json_string, string path)

返回值: string

说明：解析json的字符串json_string,返回path指定的内容。如果输入的json字符串无效，那么返回NULL。

```
hive> select get_json_object('{"store":
> {"fruit":\["weight":8,"type":"apple"}, {"weight":9,"type":"pear"}],
> "bicycle":{"price":19.95,"color":"red"}
> },
> "email":"amy@only_for_json_udf_test.net",
> "owner":"amy"
> }
> ', '$.owner') from mytable;
amy
```

16、空格字符串函数：space

语法: space(int n)

返回值: string

说明：返回长度为n的字符串

```
hive> select space(10) from mytable;
hive> select length(space(10)) from mytable;
10
```

17、重复字符串函数：repeat

语法: repeat(string str, int n)

返回值: string

说明：返回重复n次后的str字符串

```
hive> select repeat('abc',5) from mytable;
abccabccabccabc
```

18、首字符ascii函数：ascii

语法: ascii(string str)

返回值: int

说明：返回字符串str第一个字符的ascii码

```
hive> select ascii('abcde') from mytable;
97
```

19、左补足函数：lpad

语法: lpad(string str, int len, string pad)

返回值: string

说明：将str用pad进行左补足到len位

```
hive> select lpad('abc',10,'td') from mytable;
tdtdtdtabc
```

注意：与GP，ORACLE不同，pad 不能默认

20、右补足函数：rpad

语法: rpad(string str, int len, string pad)

返回值: string

说明：将str进行用pad进行右补足到len位

```
hive> select rpad('abc',10,'td') from mytable;  
abctdtdtdt
```

21、分割字符串函数: split

语法: split(string str, string pat)

返回值: array

说明: 按照pat字符串分割str，会返回分割后的字符串数组

```
hive> select split('abctdef','t') from mytable;  
["ab","cd","ef"]
```

22、集合查找函数: find_in_set

语法: find_in_set(string str, string strList)

返回值: int

说明: 返回str在strlist第一次出现的位置，strlist是用逗号分割的字符串。如果没有找该str字符，则返回0

```
hive> select find_in_set('ab','ef,ab,de') from mytable;  
2  
hive> select find_in_set('at','ef,ab,de') from mytable;  
0
```

集合统计函数

1、个数统计函数: count

语法: count(), count(expr), count(DISTINCT expr[, expr_.])

返回值: int

说明: count()统计检索出的行的个数，包括NULL值的行；count(expr)返回指定字段的非空值的个数；count(DISTINCT expr[, expr_.])返回指定字段的不同的非空值的个数

```
hive> select count(*) from mytable;  
20  
hive> select count(distinct t) from mytable;  
10
```

2、总和统计函数: sum

语法: sum(col), sum(DISTINCT col)

返回值: double

说明: sum(col)统计结果集中col的相加的结果 ; sum(DISTINCT col)统计结果中col不同值相加的结果

```
hive> select sum(t) from mytable;
100
hive> select sum(distinct t) from mytable;
70
```

3、平均值统计函数: avg

语法: avg(col), avg(DISTINCT col)

返回值: double

说明: avg(col)统计结果集中col的平均值 ; avg(DISTINCT col)统计结果中col不同值相加的平均值

```
hive> select avg(t) from mytable;
50
hive> select avg (distinct t) from mytable;
30
```

4、最小值统计函数: min

语法: min(col)

返回值: double

说明: 统计结果集中col字段的最小值

```
hive> select min(t) from mytable;
20
```

5、最大值统计函数: max

语法: maxcol)

返回值: double

说明: 统计结果集中col字段的最大值

```
hive> select max(t) from mytable;  
120
```

6、非空集合总体变量函数: var_pop

语法: var_pop(col)

返回值: double

说明: 统计结果集中col非空集合的总体变量 (忽略null)

7、非空集合样本变量函数: var_samp

语法: var_samp (col)

返回值: double

说明: 统计结果集中col非空集合的样本变量 (忽略null)

8、总体标准偏离函数: stddev_pop

语法: stddev_pop(col)

返回值: double

说明: 该函数计算总体标准偏离, 并返回总体变量的平方根, 其返回值与VAR_POP函数的平方根相同

9、样本标准偏离函数: stddev_samp

语法: stddev_samp (col)

返回值: double

说明: 该函数计算样本标准偏离

10 . 中位数函数: percentile

语法: percentile(BIGINT col, p)

返回值: double

说明: 求准确的第pth个百分位数, p必须介于0和1之间, 但是col字段目前只支持整数, 不支持浮点数类型

11、中位数函数: percentile

语法: percentile(BIGINT col, array(p1 [, p2]...))

返回值: array

说明: 功能和上述类似, 之后后面可以输入多个百分位数, 返回类型也为array, 其中为对应的百分位数。

```
select percentile(score,&lt;0.2,0.4>) from mytable;
```

取0.2 , 0.4位置的数据

12、近似中位数函数: percentile_approx

语法: percentile_approx(DOUBLE col, p [, B])

返回值: double

说明: 求近似的第pth个百分位数, p必须介于0和1之间, 返回类型为double, 但是col字段支持浮点类型。参数B控制内存消耗的近似精度, B越大, 结果的准确度越高。默认为10,000。当col字段中的distinct值的个数小于B时, 结果为准确的百分位数

13、近似中位数函数: percentile_approx

语法: percentile_approx(DOUBLE col, array(p1 [, p2]...) [, B])

返回值: array

说明: 功能和上述类似, 之后后面可以输入多个百分位数, 返回类型也为array, 其中为对应的百分位数。

14、直方图: histogram_numeric

语法: histogram_numeric(col, b)

返回值: array<struct {‘x’,‘y’}>

说明: 以b为基准计算col的直方图信息。

```
hive> select histogram_numeric(100,5) from mytable;  
[{"x":100.0,"y":1.0}]
```

复合类型构建操作

1、Map类型构建: map

语法: map (key1, value1, key2, value2, ...)

说明: 根据输入的key和value对构建map类型

```
hive> Create table mytable as select map('100','tom','200','mary') as t from  
mytable;  
hive> describe mytable;  
t          map<string ,string>  
hive> select t from mytable;  
{"100":"tom","200":"mary"}
```

2、Struct类型构建: struct

语法: struct(val1, val2, val3, ...)

说明：根据输入的参数构建结构体struct类型

```
hive> create table mytable as select struct('tom','mary','tim') as t from mytable;
hive> describe mytable;
t          struct<col1:string ,col2:string,col3:string>
hive> select t from mytable;
{"col1":"tom","col2":"mary","col3":"tim"}
```

3、array类型构建: array

语法: array(val1, val2, ...)

说明：根据输入的参数构建数组array类型

```
hive> create table mytable as select array("tom","mary","tim") as t from mytable;
hive> describe mytable;
t          array<string>
hive> select t from mytable;
["tom","mary","tim"]
```

复杂类型访问操作

1、array类型访问: A[n]

语法: A[n]

操作类型: A为array类型，n为int类型

说明：返回数组A中的第n个变量值。数组的起始下标为0。比如，A是个值为['foo', 'bar']的数组类型，那么A[0]将返回'foo',而A[1]将返回'bar'

```
hive> create table mytable as select array("tom","mary","tim") as t from mytable;
hive> select t[0],t[1],t[2] from mytable;
tom      mary      tim
```

2、map类型访问: M[key]

语法: M[key]

操作类型: M为map类型，key为map中的key值

说明：返回map类型M中，key值为指定值的value值。比如，M是值为{'f' -> 'foo', 'b' -> 'bar', 'all' -> 'foobar'}的map类型，那么M['all']将会返回'foobar'

```
hive> Create table mytable as select map('100','tom','200','mary') as t from mytable;
hive> select t['200'],t['100'] from mytable;
mary      tom
```

3、struct类型访问: S.x

语法: S.x

操作类型: S为struct类型

说明：返回结构体S中的x字段。比如，对于结构体struct foobar {int foo, int bar}，foobar.foo返回结构体中的foo字段

```
hive> create table mytable as select struct('tom','mary','tim') as t from mytable;
hive> describe mytable;
t      struct<col1:string,col2:string,col3:string>
hive> select t.col1,t.col3 from mytable;
tom     tim
```

复杂类型长度统计函数

1.Map类型长度函数: size(Map<k .V>)

语法: size(Map<k .V>)

返回值: int

说明: 返回map类型的长度

```
hive> select size(map('100','tom','101','mary')) from mytable;
2
```

2.array类型长度函数: size(Array)

语法: size(Array)

返回值: int

说明: 返回array类型的长度

```
hive> select size(array('100','101','102','103')) from mytable;
4
```

3.类型转换函数

类型转换函数: cast

语法: cast(expr as)

返回值: Expected "=" to follow "type"

说明: 返回转换后的数据类型

```
hive> select cast(1 as bigint) from mytable;  
1
```