尚硅谷大数据项目之 Hive 常用函数大全

（作者：尚硅谷研究院）

版本：V1.0

第 1 章 算数运算

1.1 加法： +

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（详见数据类型的继承关系）。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 iteblog;
10
hive> create table iteblog as select 1 + 1.2 from iteblog;
hive> describe iteblog;
_c0    double
```

1.2 减法： -

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（详见数据类型的继承关系）。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 10 - 5 from iteblog;
5
hive> create table iteblog as select 5.6 - 4 from iteblog;
hive> describe iteblog;
_c0    double
```

1.3 乘法： *

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（详见数据类型的继承关系）。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 iteblog;
200
```

1.4 除法： /

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 iteblog;
8.0
1
2
```

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 iteblog limit 1;
结果为 4
hive>select ceil(28.0/6.99999999999999) from iteblog limit 1;
结果为 5
```

1.5 取余： %

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（详见数据类型的继承关系）。

```
hive> select 41 % 5 from iteblog;
```

```
1
hive> select 8.4 % 4 from iteblog;
0.40000000000000036
```

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 iteblog;
0.4
```

1.6 位与： &

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（详见数据类型的继承关系）。

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

1.7 位或： |

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 的类型和 B 的类型的最小父类型（详见数据类型的继承关系）。

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

1.8 位异或： ^

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 的类型和 B 的类型的最小父类型（详见数据类型的继承关系）。

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

1.9 位取反： ~

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 iteblog;
-7
hive> select ~4 from iteblog;
-5
```

第 2 章 关系运算

2.1 等值比较： =

语法：A=B

操作类型：所有基本类型

说明： 如果表达式 A 与表达式 B 相等，则为 TRUE；否则为 FALSE

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

2.2 不等值比较： <>

语法： A <> B

操作类型： 所有基本类型

说明： 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 iteblog where 1 <> 2;
1
```

2.3 小于比较： <

语法： A < B

操作类型：所有基本类型

说明： 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 iteblog where 1 < 2;
1
```

2.4 小于等于比较： <=

语法： A <= B

操作类型： 所有基本类型

说明： 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 iteblog where 1 < = 1;
1
```

2.5 大于比较： >

语法： A > B

操作类型： 所有基本类型

说明： 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 iteblog where 2 > 1;
1
```

2.6 大于等于比较： >=

语法： A >= B

操作类型： 所有基本类型

说明： 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 iteblog 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 * from iteblog;
OK
2011111209 00: 00: 00     2011111209
hive> select a, b, a<b, a>b, a=b from iteblog;
2011111209 00: 00: 00     2011111209      false   true    false
```

2.7 空值判断： IS NULL

语法： A IS NULL

操作类型： 所有类型

说明： 如果表达式 A 的值为 NULL，则为 TRUE；否则为 FALSE

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

2.8 非空判断： IS NOT NULL

语法： A IS NOT NULL

操作类型： 所有类型

说明： 如果表达式 A 的值为 NULL，则为 FALSE；否则为 TRUE

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

2.9 LIKE 比较： LIKE

语法： A LIKE B

操作类型： strings

说明： 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 iteblog where 'football' like 'foot%';
1
hive> select 1 from iteblog where 'football' like 'foot____';
1
<strong>注意：否定比较时候用 NOT A LIKE B</strong>
hive> select 1 from iteblog where NOT 'football' like 'fff%';
```

```
1
```

2.10 JAVA 的 LIKE 操作： RLIKE

语法： A RLIKE B

操作类型： strings

说明：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 iteblog where 'footbar’ rlike '^f.*r$’;
1
注意：判断一个字符串是否全为数字：
hive>select 1 from iteblog where '123456' rlike '^\\d+$';
1
hive> select 1 from iteblog where '123456aa' rlike '^\\d+$';
```

2.11 REGEXP 操作： REGEXP

语法： A REGEXP B

操作类型： strings

说明： 功能与 RLIKE 相同

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

第 3 章 数值函数

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

3.2 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 iteblog;
3.1416
```

3.3 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 iteblog;
3
hive> select floor(25) from iteblog;
25
```

3.4 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 iteblog;
4
hive> select ceil(46) from iteblog;
```

```
46
```

3.5 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 iteblog;
4
hive> select ceiling(46) from iteblog;
46
```

3.6 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 iteblog;
0.6638336467363424
hive> select rand(100) from iteblog;
0.7220096548596434
hive> select rand(100) from iteblog;
0.7220096548596434
```

3.7 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 iteblog;
7.38905609893065
<strong>自然对数函数</strong>: ln
<strong>语法</strong>: ln(double a)
<strong>返回值</strong>: double
<strong>说明</strong>: 返回 a 的自然对数
1
hive> select ln(7.38905609893065) from iteblog;
2.0
```

3.8 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 iteblog;
2.0
```

3.9 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 iteblog;
3.0
```

3.10 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 iteblog;
4.0
```

3.11 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 iteblog;
16.0
```

3.12 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 iteblog;
16.0
```

3.13 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 iteblog;
4.0
```

3.14 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 iteblog;
111
```

3.15 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 iteblog;
11
hive> select hex('abc') from iteblog;
616263
```

3.16 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 iteblog;
3.9
hive> select abs(10.9) from iteblog;
10.9
```

3.17 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 iteblog;
abc
hive> select unhex('11') from iteblog;
-
```

```
hive> select unhex(616263) from iteblog;
abc
```

3.18 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 iteblog;
11
hive> select conv(17,10,2) from iteblog;
10001
```

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

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 iteblog;
0.7173560908995228
```

3.21 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 iteblog;
0.8
```

3.22 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 iteblog;
0.6216099682706644
```

3.23 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 iteblog;
0.9
```

3.24 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 iteblog;
```

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

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

3.26 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，a 可为小数

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

3.27 正切函数：tan

语法： tan(double a)

返回值： double

说明： 返回 a 的正切值

```
hive> select tan(0.8);
1.0296385570503641
```

3.28 反正切函数：atan

语法： atan(double a)

返回值： double

说明： 返回 a 的反正切值

```
hive> select atan(1.0296385570503641);
0.8
```

3.29 弧度值转换角度值：degrees

语法： degrees(double a)

返回值： double

说明： 返回 a 的角度值

```
hive> select degrees(1);
57.29577951308232
```

3.30 角度值转换成弧度值：radians

语法： radians(double a)

返回值： double

说明： 返回 a 的弧度值

```
hive> select radians(57.29577951308232);
1.0
```

3.31 判断正负函数：sign

语法： sign(double a)

返回值： double

说明： 如果 a 是正数则返回 1.0，是负数则返回-1.0，否则返回 0.0

```
hive> select sign(-4);
-1.0
```

3.32 数学 e 函数：e

语法： e()

返回值： double

说明： 数学常数 e

```
hive> select e();
2.718281828459045
```

3.33 数学 pi 函数：pi

语法： pi()

返回值： double

说明： 圆周率 π

```
hive> select pi();
3.141592653589793
```

3.34 阶乘函数：factorial

语法： factorial(int a)

返回值： bigint

说明： 求 a 的阶乘

```
hive> select factorial(5);
120
```

3.35 立方根函数：cbrt

语法： cbrt(double a)

返回值： double

说明： 求 a 的立方根

```
hive> select cbrt(27);
3
```

3.36 左移函数：shiftleft

语法： shiftleft(BIGINT a, int b)

返回值： int bigint

说明： 按位左移

```
hive> select shiftleft(4,2);
16
```

3.37 右移函数：shiftright

语法： shiftright(BIGINT a, int b)

返回值： int bigint

说明： 按位右移

```
hive> select shiftright(16,1);
 8
```

3.38 无符号按位右移函数：shiftrightunsigned

语法： shiftrightunsigned(BIGINT a, int b)

返回值： int bigint

说明： 无符号按位右移（<<<）

```
hive> select shiftrightunsigned(32,2)
8
```

3.39 求最大值函数：greatest

语法： greatest(T v1, T v2, …)

返回值： T

说明： 求最大值

```
hive> select greatest(1,2,3);
3
```

3.40 求最小值函数：least

语法： least(T v1, T v2, …)

返回值： T

说明： 求最小值

```
hive> select least(1,2,3);
1
```

3.41 银行家舍入法函数：bround

语法： bround(double a)

返回值： double

说明： 银行家舍入法（1-4：舍，6-9：进，5->前位数是偶：舍，5->前位数是奇：进）

```
hive> select bround(3.5)
3.0
```

3.42 银行家精确舍入法函数：bround

语法： bround(double a，int d)

返回值： double

说明： 银行家舍入法,保留 d 位小数

```
hive> select bround(3.15，1)
3.1
hive> select bround(3.25，1)
3.3
```

第 4 章 日期函数

4.1 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 iteblog;
20111208
```

4.2 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 iteblog;
1323309615
```

4.3 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 iteblog;
1323234063
```

4.4 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 iteblog;
1323234063
```

4.5 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 iteblog;
2011-12-08
```

4.6 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 iteblog;
2011
hive> select year('2012-12-08') from iteblog;
2012
```

4.7 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 iteblog;
12
hive> select month('2011-08-08') from iteblog;
8
```

4.8 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 iteblog;
8
hive> select day('2011-12-24') from iteblog;
24
```

4.9 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 iteblog;
10
```

4.10 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 iteblog;
3
```

4.11 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 iteblog;
1
```

4.12 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 iteblog;
```

```
49
```

4.13 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 iteblog;
213
```

4.14 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 iteblog;
2012-12-18
```

4.15 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 iteblog;
2012-11-28
```

4.16 转化成指定的时区下时间戳函数： from_utc_timestamp

语法： from_utc_timestamp(timestamp, string timezone)

返回值： timestamp

说明： 如果给定的时间戳并非 UTC，则将其转化成指定的时区下时间戳

```
hive> select from_utc_timestamp( ‘1970-01-01 08: 00: 00’ , ‘PST’ );
1970-01-01 00: 00: 00
```

4.17 转化成 UTC 下的时间戳函数： to_utc_timestamp

语法： to_utc_timestamp(timestamp, string timezone)

返回值： timestamp

说明： 如果给定的时间戳指定的时区下时间戳，则将其转化成 UTC 下的时间戳。

```
hive> select to_utc_timestamp( ‘1970-01-01 00: 00: 00’ , ‘PST’ );
1970-01-01 08: 00: 00
```

4.18 当前时间日期函数：current_date

语法： current_date()

返回值： date

说明： 返回当前时间日期

```
hive> select current_date;
2022-01-06
```

4.19 当前时间日期函数：current_timestamp

语法： current_timestamp()

返回值： timestamp

说明： 返回当前时间戳

```
hive> select current_timestamp();
2022-01-06 22: 52: 11.309
```

4.20 月份增加函数：add_months

语法： add_months(string start_date, int num_months)

返回值： string

说明： 返回当前时间下再增加 num_months 个月的日期

```
hive> select add_months('1996-10-21',10);
1997-08-21
```

4.21 最后一天的日期函数：last_day

语法： last_day(string date)

返回值： string

说明： 返回这个月的最后一天的日期，忽略时分秒部分（HH：mm：ss）

```
hive> select last_day(current_date());
2020-07-31
```

4.22 下一个星期 X 所对应的日期函数：next_day

语法： next_day(string start_date, string day_of_week)

返回值： string

说明： 返回当前时间的下一个星期 X 所对应的日期 如：next_day(“2015-01-14”, “TU”) = 2015-01-20 以 2015-01-14 为开始时间，其下一个星期二所对应的日期为 2015-01-20

```
hive> select next_day(current_date(),'su');
2020-07-19
```

4.23 时间的最开始年份或月份函数：trunc

语法： trunc(string date, string format)

返回值： string

说明： 返回时间的最开始年份或月份 如 trunc(“2016-06-26”,“MM”)=2016-06-01 trunc(“2016-06-26”,“YY”)=2016-01-01 注意所支持的格式为 MONTH/MON/MM, YEAR/YYYY/YY

```
hive> select trunc(current_date(),'MM');
2020-07-01
```

4.24 相差的月份函数：months_between

语法： months_between(date1, date2)

返回值： double

说明：返回 date1 与 date2 之间相差的月份，如 date1>date2，则返回正，如果 date1<date2,则返回负，否则返回 0.0 如：months_between(‘1997-02-28 10：30：00’, ‘1996-10-30’) = 3.94959677 1997-02-28 10：30：00 与 1996-10-30 相差 3.94959677 个月

```
hive> select months_between(current_date(),'2020-5-13');
2.0
```

4.25 指定格式返回时间函数：date_format

语法： date_format(date/timestamp/string ts, string fmt)

返回值： string

说明： 按指定格式返回时间 date 如：date_format(“2016-06-22”,“MM-dd”)=06-22

```
hive> select date_format(current_date(),'MM.dd');
07.13
```

4.26 当前星期函数：dayofweek

语法： dayofweek(date)

返回值： int

说明： 返回日期那天的周几

```
hive> select dayofweek(current_date());
2
```

4.27 季节函数：quarter

语法： quarter(date/timestamp/string)

返回值： int

说明： 返回当前时间属性哪个季度 如 quarter(“2015-04-08”) = 2

第 5 章 条件函数

5.1 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（valueTrue，valueFalseOrNull 为泛型）

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

5.2 空查找函数： nvl

语法： nvl(T value, T default_value)

返回值： T

说明： 如果 value 值为 NULL 就返回 default_value,否则返回 value

```
hive> select nvl(null,5);
5
```

5.3 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,44,55);
 44
```

5.4 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 4 WHEN 5 THEN 5 WHEN 4 THEN 4 ELSE 3 END;
4
```

5.5 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 5>0 THEN 5 WHEN 4>0 THEN 4 ELSE 0 END;
5
```

5.6 空值判断函数：isnull

语法： isnull(a)

返回值： boolean

说明： 如果 a 为 null 就返回 true，否则返回 false

```
hive> select isnull(5);
false
```

5.7 非空值判断函数：isnotnull

语法： isnotnull (a)

返回值： boolean

说明： 如果 a 为非 null 就返回 true，否则返回 false

```
hive> select isnotnull(5);
true
```

第 6 章 字符串函数

6.1 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('abcedfg') from iteblog;
7
```

6.2 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(abcedfg’) from iteblog;
gfdecba
```

6.3 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 iteblog;
abcdefgh
```

6.4 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 iteblog;
abc,def,gh
```

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

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

6.7 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 iteblog;
ABSED
hive> select ucase('abSEd') from iteblog;
ABSED
```

6.8 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 iteblog;
absed
hive> select lcase('abSEd') from iteblog;
absed
```

6.9 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 iteblog;
abc
```

6.10 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 iteblog;
abc
```

6.11 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 iteblog;
abc
```

6.12 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 iteblog;
fb
```

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

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

6.15 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 iteblog;
amy
```

6.16 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 iteblog;
hive> select length(space(10)) from iteblog;
10
```

6.17 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 iteblog;
abcabcabcabcabc
```

6.18 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 iteblog;
97
```

6.19 左补足函数：lpad

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

返回值： string

说明：将 str 进行用 pad 进行左补足到 len 位

```
hive> select lpad('abc',10,'td') from iteblog;
tdtdtdtabc
注意：与 GP，ORACLE 不同，pad 不能默认
```

6.20 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 iteblog;
abctdtdtdt
```

6.21 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('abtcdtef','t') from iteblog;
["ab","cd","ef"]
```

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

6.23 转换成 64 位的字符串：base64

语法： base64(binary bin)

返回值： string

说明：将二进制 bin 转换成 64 位的字符串

6.24 字符串连接函数：context_ngrams

语法：context_ngrams(array, array, int K, int pf)

返回值： array<struct<string,double>>

说明：与 ngram 类似，但 context_ngram()允许你预算指定上下文(数组)来去查找子序列，具体看 StatisticsAndDataMining(这里的解释更易懂)

6.25 将数值 X 转换成"#,###,###.##"格式字符串：format_number

语法： format_number(number x, int d)

返回值： string

说明：将数值 X 转换成"#,###,###.##"格式字符串，并保留 d 位小数，如果 d 为 0，将进行四舍五入且不保留小数

```
hive> select format_number(123345.65545,2);
123,345.66
```

6.26 指定的字符集将二进制值 bin 解码成字符串：decode

语法： decode(binary bin, string charset)

返回值： string

说明：使用指定的字符集 charset 将二进制值 bin 解码成字符串，支持的字符集有：‘US-ASCII’，‘ISO-8859-1’，‘UTF-8’，‘UTF-16BE’，‘UTF-16LE’，‘UTF-16’，如果任意输入参数为 NULL 都将返回 NULL

6.27 指定的字符集 charset 将字符串编码成二进制值：encode

语法： encode(string src, string charset)

返回值： binary

说明：使用指定的字符集 charset 将字符串编码成二进制值，支持的字符集有：‘US-ASCII’，‘ISO-8859-1’，‘UTF-8’，‘UTF-16BE’，‘UTF-16LE’，‘UTF-16’，如果任一输入参数为 NULL 都将返回 NULL

6.28 文件数据与字符串 str 匹配： in_file

语法： in_file(string str, string filename)

返回值： boolean

说明：如果文件名为 filename 的文件中有一行数据与字符串 str 匹配成功就返回 true

6.29 查找字符串 str 中子字符串 substr 出现的位置：instr

语法： instr(string str, string substr)

返回值： int

说明：查找字符串 str 中子字符串 substr 出现的位置，如果查找失败将返回 0，如果任一参数为 Null 将返回 null，注意位置为从 1 开始的

```
hive> select instr(‘dvfgefggdgaa’,‘aa’);
11
```

6.30 第一次出现的位置：locate

语法： locate(string substr, string str[, int pos])

返回值： int

说明：查找字符串 str 中的 pos 位置后字符串 substr 第一次出现的位置

```
hive> select locate(‘aa’,‘aabbedfaad’,2);
8
```

6.31 返回出现次数 TOP K 的的子序列：ngrams

语法： ngrams(array, int N, int K, int pf)

返回值： array<struct<string,double>>

说明：返回出现次数 TOP K 的的子序列,n 表示子序列的长度，具体看 StatisticsAndDataMining (这里的解释更易懂)

6.32 printf 风格格式输出字符串：printf

语法：printf(String format, Obj… args)

返回值： string

说明：按照 printf 风格格式输出字符串

```
hive> select printf(‘abfhg’);
Abfhg
```

6.33 字符串 str 将被转换成单词数组：sentences

语法： sentences(string str, string lang, string locale)

返回值： array

说明：字符串 str 将被转换成单词数组，如：sentences(‘Hello there! How are you?’) =((“Hello”, “there”), (“How”, “are”, “you”))

```
hive> select sentences(‘Hello there! How are you?’);
[[“Hello”,“there”],[“How”,“are”,“you”]]
```

6.34 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(‘abc’);
cba
```

6.35 字符串 str 按照指定分隔符转换成 Map： split

语法：str_to_map(text[, delimiter1, delimiter2])

返回值： map<string,string>

说明：将字符串 str 按照指定分隔符转换成 Map，第一个参数是需要转换字符串，第二个参数是键值对之间的分隔符，默认为逗号;第三个参数是键值之间的分隔符，默认为"="

6.36 截取第 count 分隔符之前的字符串：substring_index

语法： substring_index(string A, string delim, int count)

返回值： string

说明：截取第 count 分隔符之前的字符串，如 count 为正则从左边开始截取，如果为负则从右边开始截取

6.37 字符串替换成 to 中的字符串：substring_index

语法： translate(string|char|varchar input, string|char|varchar from, string|char|varchar to)

返回值： string

说明：将 input 出现在 from 中的字符串替换成 to 中的字符串 如：translate(“MOBIN”,“BIN”,“M”)=“MOM”

```
hive> select translate(“MOBIN”,“BIN”,“M”);
MOM
```

6.38 首字母大写函数：initcap

语法：initcap(string A)

返回值： string

说明：将字符串 A 转换第一个字母大写其余字母的字符串

```
hive> select initcap(‘abcd def’);
Abcd Def
```

6.39 两个字符串之间的差异大小： levenshtein

语法： levenshtein(string A, string B)

返回值： int

说明：计算两个字符串之间的差异大小 如：levenshtein(‘kitten’, ‘sitting’) = 3

```
hive> select levenshtein(‘kitten’, ‘sitting’);
3
```

6.40 字符串转换成 soundex 字符串：soundex

语法： soundex(string A)

返回值： string

说明： 将普通字符串转换成 soundex 字符串

第 7 章 聚合函数

7.1 个数统计函数： count

语法： count(*), count(expr), count(DISTINCT expr[, expr…])

返回值： bigint

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(DISTINCTexpr[, expr_.])统计提供非 NULL 且去重后的 expr 表达式值的行数

7.2 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

7.3 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

7.4 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

7.5 最大值统计函数： max

语法：max(col)

返回值： double

说明：统计结果集中 col 字段的最大值

7.6 非空集合总体变量函数：var_pop

语法： variance(col), 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），（求指定列数值的方差）

7.7 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）（求指定列数值的样本方差）

7.8 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（求指定列数值的标准偏差）

7.9 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，（求指定列数值的样本标准偏差）

7.10 协方差函数：covar_pop

语法： covar_pop(col1, col2)

返回值： double

说明：求指定列数值的协方差

7.11 样本协方差函数：covar_samp

语法： covar_samp(col1, col2)

返回值： double

说明：求指定列数值的样本协方差

7.12 相关系数函数：corr

语法：corr(col1, col2)

返回值： double

说明：返回两列数值的相关系数

7.13 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

7.14 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,<0.2,0.4>) from lxw_dual；取 0.2，0.4 位置的数据
1
```

7.15 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

7.16 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

7.17 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)
[{“x”:100.0,“y”:1.0}]
```

7.18 高级聚合：collect_list/collect_set

1）collect_list 收集并形成 list 集合，结果不去重

语法：collect_list(col)

返回值：array

说明：将某分组内该字段的所有值收集成为一个数组，结果不去重

```
hive>
select
  sex,
  collect_list(job)
from
  employee
group by
  sex
```

结果：

```
女 ["行政","研发","行政","前台"]
男 ["销售","研发","销售","前台"]
```

2）collect_set 收集并形成 set 集合，结果去重

语法：collect_set(col)

返回值：array

说明：将某分组内该字段的所有值收集成为一个数组，结果去重

```
hive>
select
  sex,
  collect_set(job)
from
  employee
group by
  sex
```

结果：

```
女 ["行政","研发","前台"]
男 ["销售","研发","前台"]
```

第 8 章 表生成函数

8.1 explode

语法： explode(array a)

返回值： Array Type

说明：对于 a 中的每个元素，将生成一行且包含该元素

8.2 explode

语法： explode(ARRAY)

返回值：N rows

说明：每行对应数组中的一个元素

8.3 explode

语法： explode(MAP)

返回值：N rows

说明：每行对应每个 map 键-值，其中一个字段是 map 的键，另一个字段是 map 的值

8.4 posexplode

语法： posexplode(ARRAY)

返回值：N rows

说明：与 explode 类似，不同的是还返回各元素在数组中的位置

8.5 posexplode

语法： stack(INT n, v_1, v_2, …, v_k)

返回值：N rows

说明：把 M 列转换成 N 行，每行有 M/N 个字段，其中 n 必须是个常数

8.6 posexplode

语法： json_tuple(jsonStr, k1, k2, …)

返回值： tuple

说明：从一个 JSON 字符串中获取多个键并作为一个元组返回，与 get_json_object 不同的是此函数能一次获取多个键值

8.7 parse_url_tuple

语法： parse_url_tuple(url, p1, p2, …)

返回值： tuple

说明：返回从 URL 中抽取指定 N 部分的内容，参数 url 是 URL 字符串，而参数 p1,p2,…是要抽取的部分，这个参数包含 HOST, PATH, QUERY, REF, PROTOCOL, AUTHORITY, FILE, USERINFO, QUERY：

8.8 parse_url_tuple

语法： inline(ARRAY<STRUCT[,STRUCT]>)

返回值： tuple

说明：将结构体数组提取出来并插入到表中

8.9 示例

一进多出（一行进入，多行输出）。

1）explode 将数组或者 map 展开

```
hive> select explode(array('a','b','d','c'));
```

结果：

```
a
b
d
c
```

2）json_tuple 取出 json 字符串中属性的值

```
hive>
select json_tuple('{"name":"王二狗","sex":"男","age":"25"}','name','sex','age');
```

结果：

```
王二狗  男  25
```

第 9 章 复合类型构建操作

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

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

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

第 10 章 复杂类型访问操作

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

10.2 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 iteblog as select map('100','tom','200','mary') as t from iteblog;
hive> select t['200'],t['100'] from iteblog;
mary    tom
```

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

第 11 章 复杂类型长度统计函数

11.1 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 iteblog;
2
```

11.2 array 类型长度函数： size(Array<T>)

语法： size(Array<T>)

返回值： int

说明： 返回 array 类型的长度

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

11.3 类型转换函数

1）转换成二进制： binary

语法： binary(string|binary)

返回值： binary

说明： 将输入的值转换成二进制

2）类型转换函数： cast

语法： cast(expr as <type>)

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 iteblog;
1
```

第 12 章 窗口函数

基本语法：函数 + over([partition by ...] [order by ...] [窗口子句])

• over 表示开窗，默认窗口大小会包含所有数据。

• partition by 表示根据字段再划分一个细窗口，相同字段进入同一个细窗口里面，每个窗口之间相互独立，窗口子句对于每个细窗口独立生效。

• order by 表示窗口内按什么排序，如果只有 over 表示直接最大窗口排序；如果有 partition by 每个细窗口单独排序。

• 窗口子句，可以进一步限定范围

(rows | range) between (unbounded | [num]) preceding and ([num] preceding | current row | (unbounded | [num]) following

(rows | range) between current row and (current row | (unbounded | [num]) following)

(rows | range) between [num] following and (unbounded | [num]) following

```
rows between unbounded preceding and unbounded following
```

行的范围为上无边界到下无边界（第一行到最后一行）。

注：窗口函数是一行一行执行的。

12.1 偏移量函数：lag

语法：lag(col,n,default_val)

返回值：字段类型

说明：往前第 n 行数据。

12.2 偏移量函数：lead

语法：lead(col,n, default_val)

返回值：字段类型

说明：往后第 n 行数据。

12.3 窗口分析函数：first_value

语法：first_value (col,true/false)

返回值：字段类型

说明：当前窗口下的第一个值，第二个参数为 true，跳过空值。

12.4 窗口分析函数：last_value

语法：last_value (col,true/false)

返回值：字段类型

说明：当前窗口下的最后一个值，第二个参数为 true，跳过空值。

12.5 跳跃排序函数：rank

语法：rank() over(……)

返回值：int

说明：排名相同时会重复，总数不会减少（12225……）。

12.6 不跳跃排序函数：dense_rank

语法：dense_rank() over(……)

返回值：int

说明：排名相同时会重复，总数会减少（12223……）。

12.7 顺序唯一的排序函数：row_number

语法：row_number() over(……)

返回值：int

说明：行号（1234567……）。

12.8 分组函数：lead

语法：ntile() over(……)

返回值：int

说明：分组并给上组号。