Hive常用函数

# 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(28.0/6.999999999999999999999) 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;  0.5577432776034763  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](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](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

1. **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。