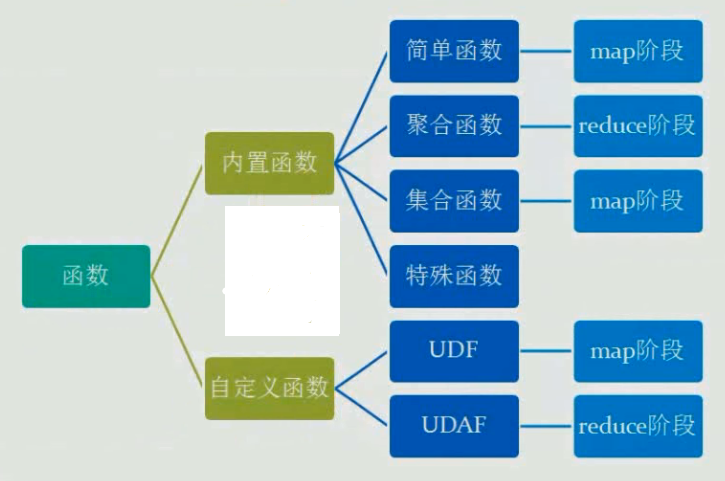
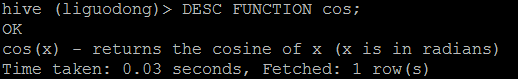
### ****函数分类****



### ****HIVE CLI命令****

显示当前会话有多少函数可用   
SHOW FUNCTIONS;

显示函数的描述信息   
DESC FUNCTION concat;



显示函数的扩展描述信息   
DESC FUNCTION EXTENDED concat;

### ****简单函数****

**函数的计算粒度为单条记录。**   
关系运算   
数学运算   
逻辑运算   
数值计算   
类型转换   
日期函数   
条件函数   
字符串函数   
统计函数

### ****聚合函数****

函数处理的数据粒度为多条记录。   
sum()—求和   
count()—求数据量   
avg()—求平均直   
distinct—求不同值数   
min—求最小值   
max—求最人值

### ****集合函数****

复合类型构建   
复杂类型访问   
复杂类型长度

### ****特殊函数****

#### ****窗口函数****

**应用场景**   
用于分区排序   
动态Group By   
Top N   
累计计算   
层次查询

**Windowing functions**

lead

lag

FIRST\_VALUE

LAST\_VALUE

#### ****分析函数****

**Analytics functions**

RANK

ROW\_NUMBER

DENSE\_RANK

CUME\_DIST

PERCENT\_RANK

NTILE

#### ****混合函数****

java\_method(class,method [,arg1 [,arg2])reflect(class,method [,arg1 [,arg2..]])hash(a1 [,a2...])

#### ****UDTF****

lateralView: LATERAL VIEW udtf(expression) tableAlias AS columnAlias (‘,‘ columnAlias)\* fromClause: FROM baseTable (lateralView)\*

ateral view用于和split, explode等UDTF一起使用，它能够将一行数据拆成多行数据，在此基础上可以对拆分后的数据进行聚合。lateral view首先为原始表的每行调用UDTF，UTDF会把一行拆分成一或者多行，lateral view再把结果组合，产生一个支持别名表的虚拟表。

### ****常用函数Demo：****

create table employee(

id string,

money double,

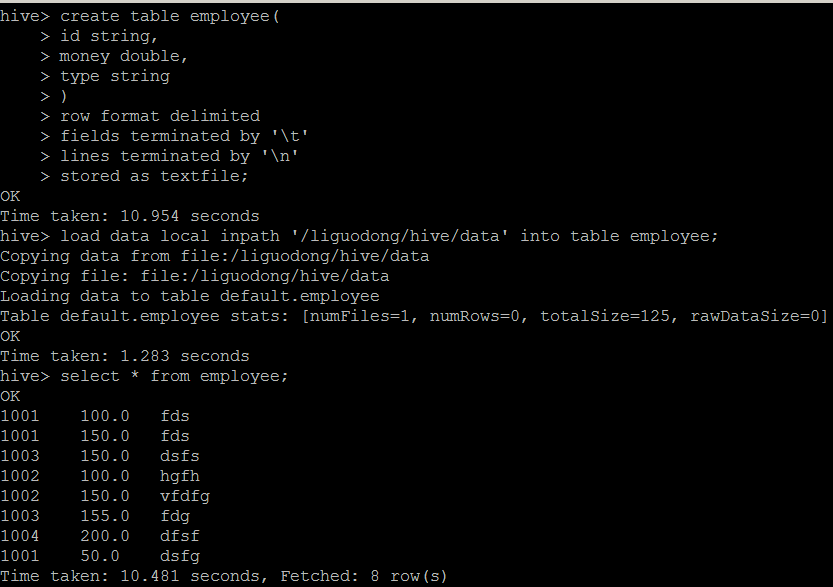
type string)row format delimited

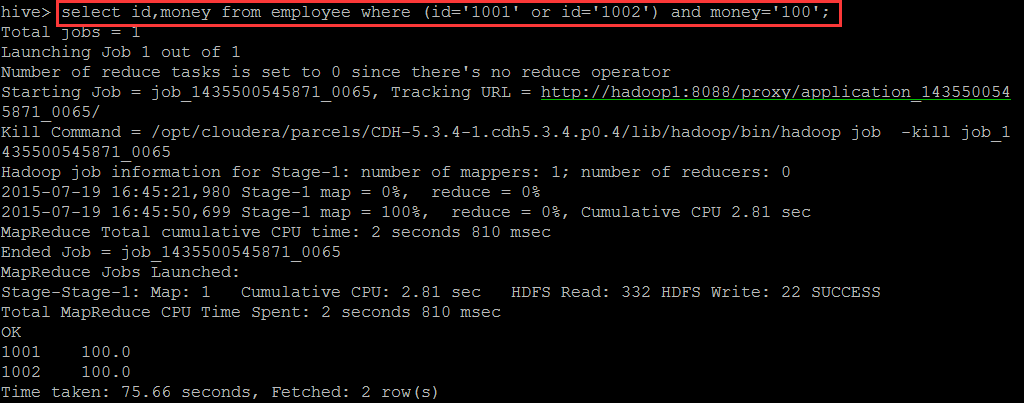
fields terminated by ‘\t‘

lines terminated by ‘\n‘

stored as textfile;load data local inpath ‘/liguodong/hive/data‘ into table employee;select \* from employee;

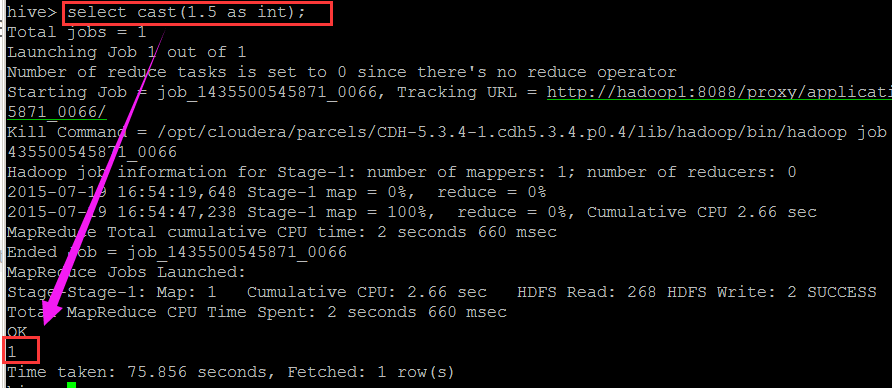
优先级依次为NOT AND ORselect id,money from employee where (id=‘1001‘ or id=‘1002‘) and money=‘100‘;

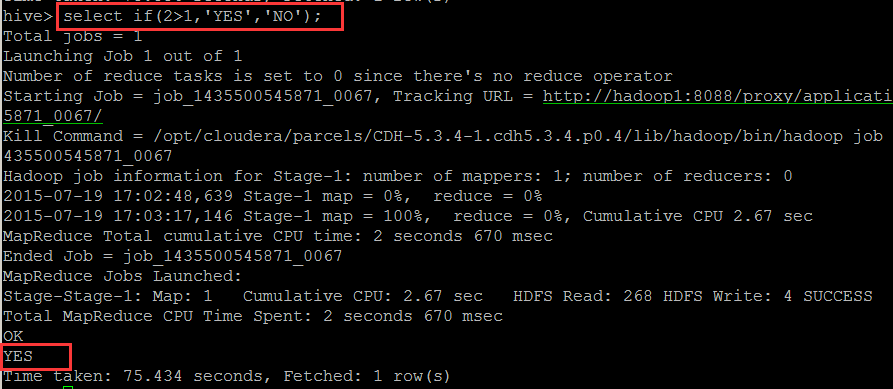




**cast类型转换**

select cast(1.5 as int);





**if判断**

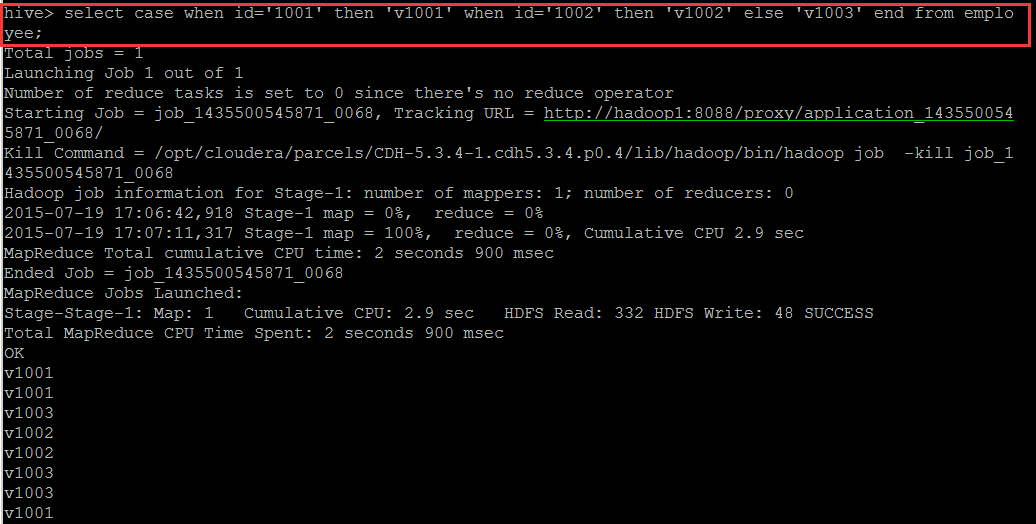
if(con,‘‘,‘‘);

hive (default)> select if(2>1,‘YES‘,‘NO‘);

YES

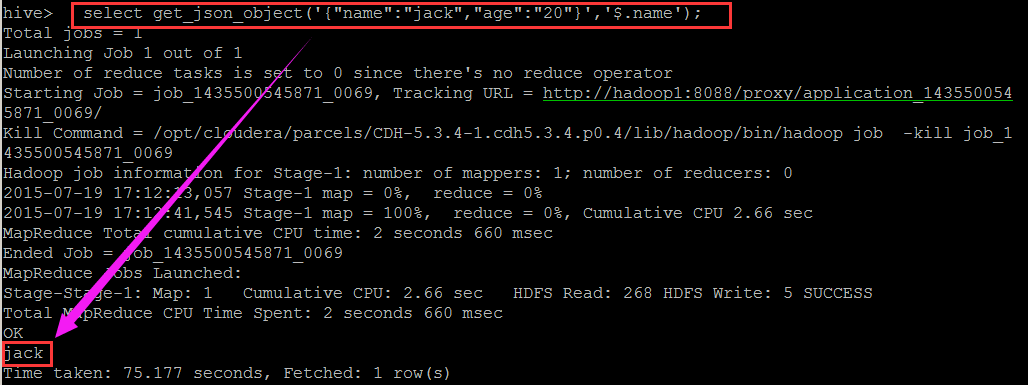
case when con then ‘‘ when con then ‘‘ else ‘‘ end (‘‘里面类型要一样)

select case when id=‘1001‘ then ‘v1001‘ when id=‘1002‘ then ‘v1002‘ else ‘v1003‘ end from employee;



get\_json\_object

get\_json\_object(json 解析函数，用来处理json，必须是json格式)select get\_json\_object(‘{"name":"jack","age":"20"}‘,‘$.name‘);

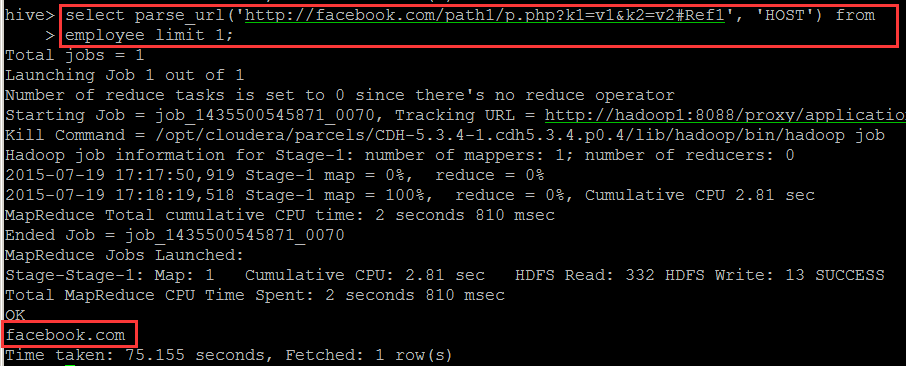


**URL解析函数**

parse\_url(string urlString, string partToExtract [, string keyToExtract])

select parse\_url(‘http://facebook.com/path1/p.php?k1=v1&k2=v2#Ref1‘, ‘HOST‘) from

employee limit 1;



**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   
**举例：**

hive> select concat(‘abc‘,‘def’,‘gh‘) from lxw\_dual;

abcdefgh

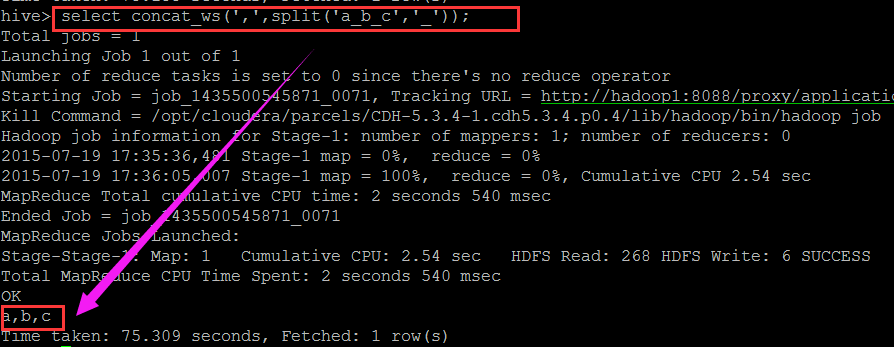
**带分隔符字符串连接函数： concat\_ws**   
语法: concat\_ws(string SEP, string A, string B…)   
返回值: string   
说明：返回输入字符串连接后的结果， SEP 表示各个字符串间的分隔符

concat\_ws(string SEP, array<string>)

举例：

hive> select concat\_ws(‘,‘,‘abc‘,‘def‘,‘gh‘) from lxw\_dual;

abc,def,gh



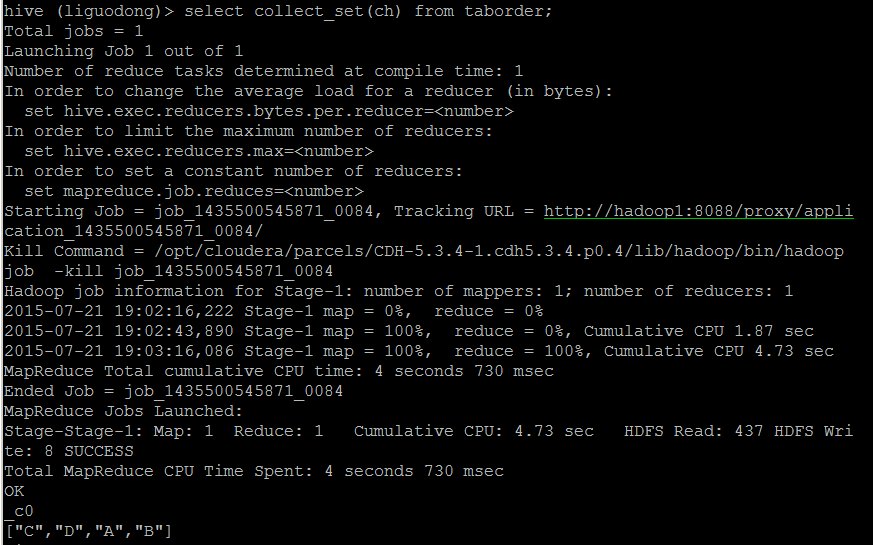
列出该字段所有不重复的值，相当于去重

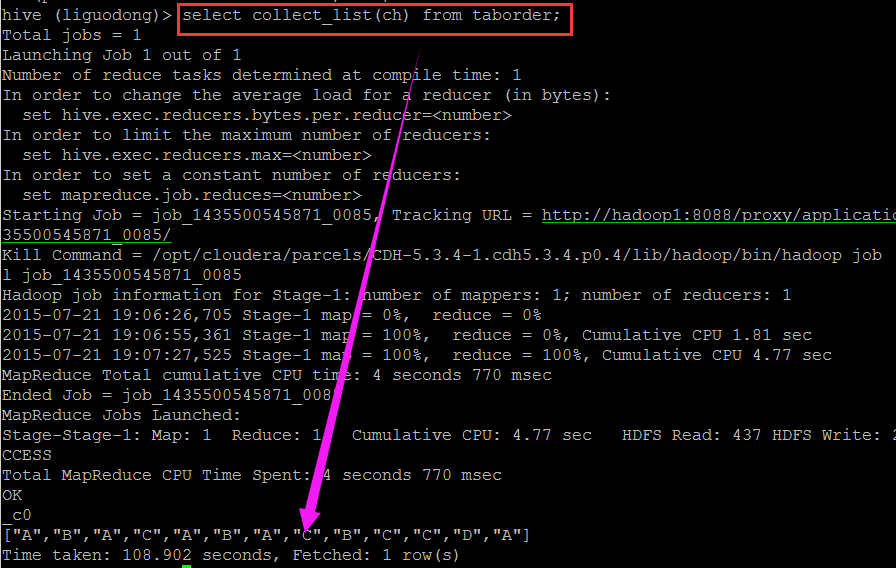
collect\_set(id) //返回的是数组

列出该字段所有的值，列出来不去重

collect\_list(id) //返回的是数组

select collect\_set(id) from taborder;





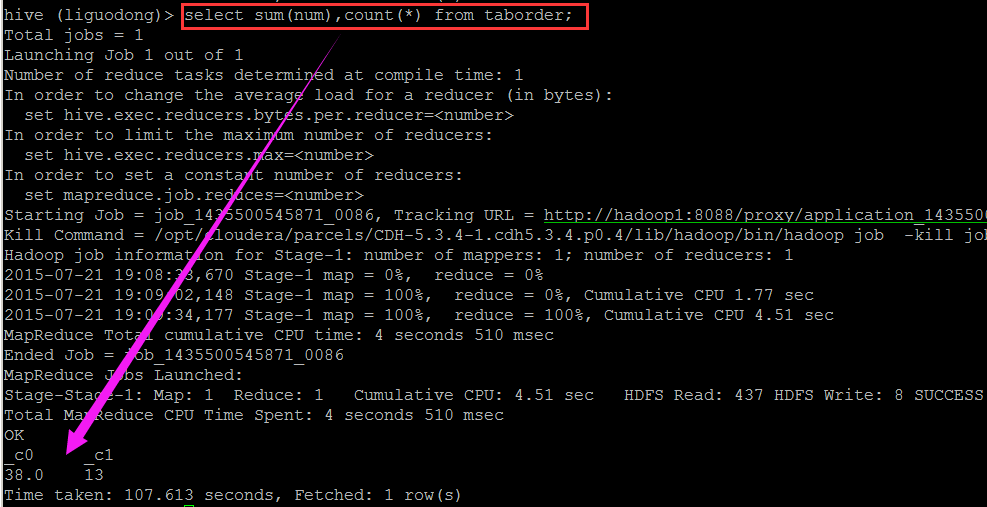
求和

sum(money)

统计列数

count(\*)

select sum(num),count(\*) from taborder;

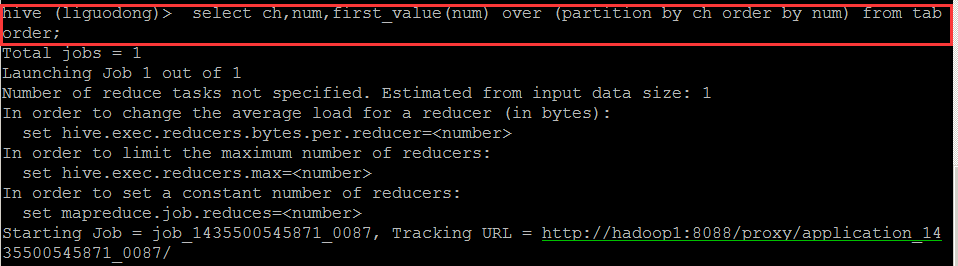


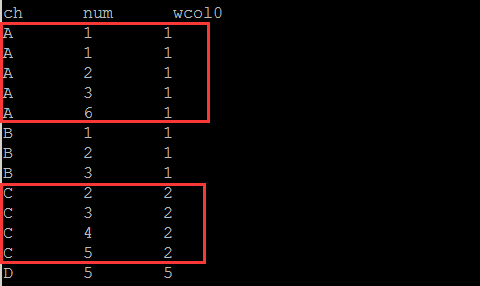
**窗口函数**

first\_value(第一行值)

first\_value(money) over (partition by id order by money)

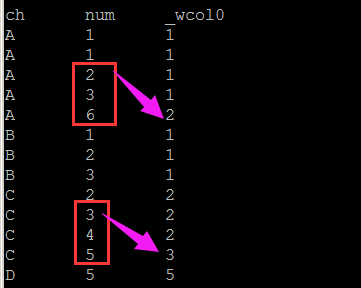
select ch,num,first\_value(num) over (partition by ch order by num) from taborder;





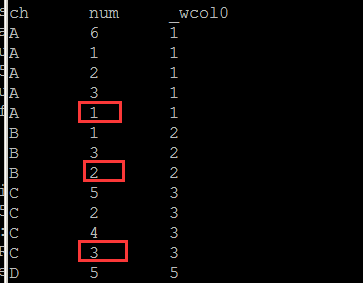
rows between 1 preceding and 1 following (当前行以及当前行的前一行与后一行)

hive (liguodong)> select ch,num,first\_value(num) over (partition by ch order by num ROWS BETWEEN 2 PRECEDING AND CURRENT ROW) from taborder;



last\_value 最后一行值

hive (liguodong)> select ch,num,last\_value(num) over (partition by ch) from taborder;



lead

去当前行后面的第二行的值

lead(money,2) over (order by money)

lag

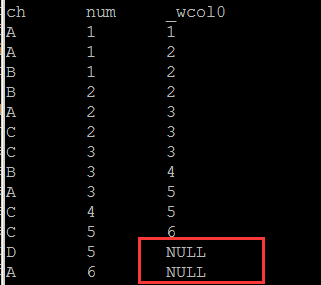
去当前行前面的第二行的值

lag(money,2) over (order by money)

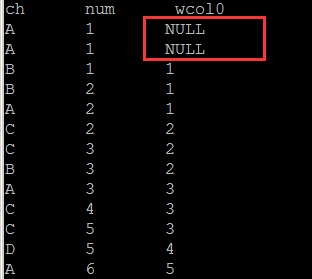
```

```

select ch, num, lead(num,2) over (order by num) from taborder;



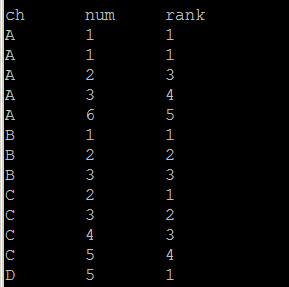
select ch, num, lag(num,2) over (order by num) from taborder;



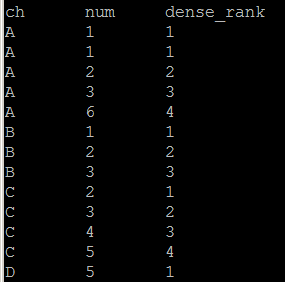
**rank排名**

rank() over(partition by id order by money)

select ch, num, rank() over(partition by ch order by num) as rank from taborder;



select ch, num, dense\_rank() over(partition by ch order by num) as dense\_rank from taborder;



**cume\_dist**

cume\_dist (相同值的最大行号/行数)

cume\_dist() over (partition by id order by money)

percent\_rank (相同值的最小行号-1)/(行数-1)

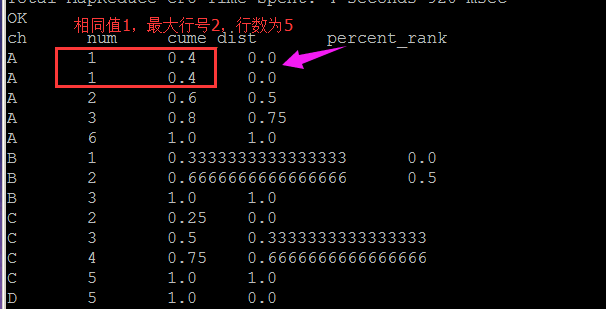
第一个总是从0开始

percent\_rank() over (partition by id order by money)

select ch,num,cume\_dist() over (partition by ch order by num) as cume\_dist,

percent\_rank() over (partition by ch order by num) as percent\_rank

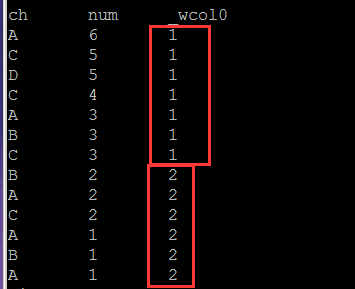
from taborder;



ntile分片

ntile(2) over (order by money desc) 分两份

select ch,num,ntile(2) over (order by num desc) from taborder;



**混合函数**

select id,java\_method("java.lang,Math","sqrt",cast(id as double)) as sqrt from hiveTest;

**UDTF**

select id,adid

from employee

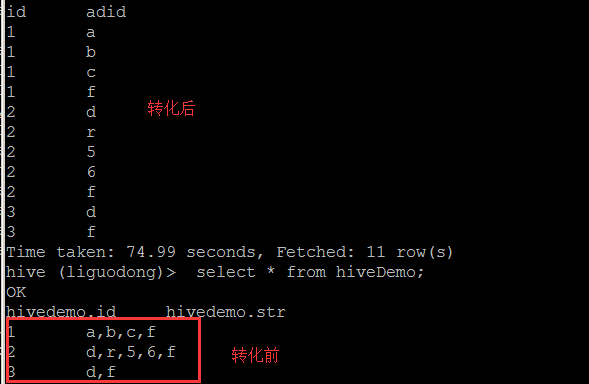
lateral view explode(split(type,‘B‘)) tt as adid;

explode 把一列转成多行

hive (liguodong)> select id,adid

> from hiveDemo

> lateral view explode(split(str,‘,‘)) tt as adid;

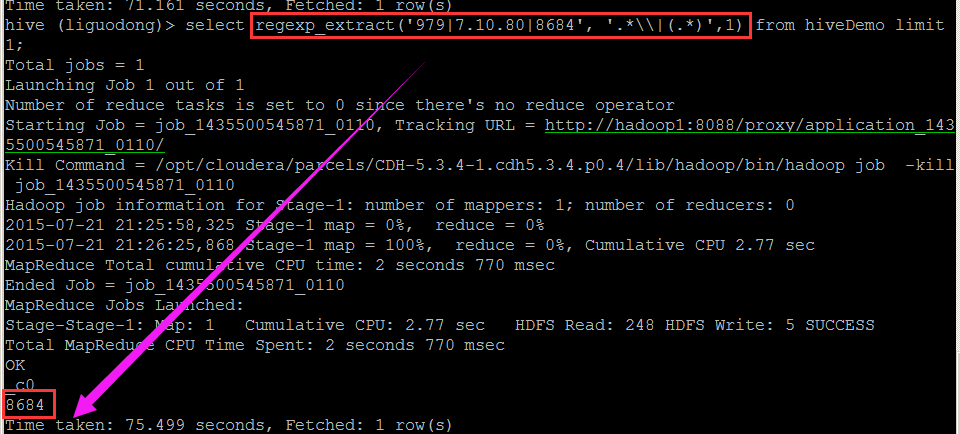


**正则表达式**   
使用正则表达式的函数   
regexp\_replace(string subject A,string B,string C)   
regexp\_extract(string subject,string pattern,int index)

hive> select regexp\_replace(‘foobar‘, ‘oo|ar‘, ‘‘) from lxw\_dual;

fb

hive> select regexp\_replace(‘979|7.10.80|8684‘, ‘.\*\\|(.\*)‘,1) from hiveDemo limit 1;



hive> select regexp\_replace(‘979|7.10.80|8684‘, ‘(.\*?)\\|(.\*)‘,1) from hiveDemo limit 1;

