

# SQL优化

2023-10-13 11:10

# 目录

## 1、Hive 里面的IF三元表达式

Tchouse 定义存储过程来优化

## 2、Hive 里面的DateDiff 函数

hive months\_between(endDate,startDate)

hive 出生年月份 判断生肖

Hive 里面decode的用法

根据身份证号码来判断性别， 年龄， 地域

Hive 里面isdate 函数

Tchouse 里面 从身份证号码获取出生日期

## 1、Hive 里面的IF三元表达式

```
1  if(princpl_overdue_starting_ddate<=interst_overdue_starting_ddate,
2      concat(substr(b.princpl_overdue_starting_ddate,1,4),'-
      ',substr(b.princpl_overdue_starting_ddate,5,2),'-
      ',substr(b.princpl_overdue_starting_ddate,7,2)),
3      concat(substr(b.interst_overdue_starting_ddate,1,4),'-
      ',substr(b.interst_overdue_starting_ddate,5,2),'-
      ',substr(b.interst_overdue_starting_ddate,7,2))
4      ))+1
```

## Tchouse 定义存储过程来优化

```
1  -- 创建pgsql 里面的存储过程
2
3  create or replace function decode(p_condtion boolean,p_first text, p_last text)
4  returns text
5  as
6  $$
7  declare v_ret text;
8  begin
9  /*
10   * 功能说明：模拟hive 里面的if(条件, value1, value2)三元表达式
11   * 参数：p_condtion 接受的条件 1==1, 1<=2 这种判断条件
12   * 实现原理：p_condtion 为真，则是返回p_first，否则返回p_last
13   * */
14
15  v_ret:=null;
16  if true = p_condtion then
17      v_ret := p_first;
18  elsif false = p_condtion then
19      v_ret := p_last;
20
21  end if;
22  return v_ret;
23  end;
24  $$
25  language plpgsql;
```

测试过程:

```
mydb=# create or replace function decode(p_condtion boolean,p_first text, p_last text)
mydb=#
mydb=# returns text
mydb=# as
mydb=# $$
mydb$$ declare v_ret text;
mydb$$ begin
mydb$$ v_ret:=null;
mydb$$ if true = p_condtion then
mydb$$ v_ret := p_first;
mydb$$ elsif false = p_condtion then
mydb$$ v_ret := p_last;
mydb$$
mydb$$ end if;
mydb$$ return v_ret;
mydb$$ end;
mydb$$ $$
mydb=# language plpgsql;
CREATE FUNCTION
mydb=# select decode(1=1, 'a','a');
decode
-----
a
(1 row)

mydb=# select decode(1=1, 'a','b');
decode
-----
a
(1 row)
```

转换之后结果:

```
1  date_part('day', a.data_date::date -
   decode(princpl_overdue_starting_ddate<=interst_overdue_starting_ddate,
2  concat(substr(b.princpl_overdue_starting_ddate,1,4),'-
   ',substr(b.princpl_overdue_starting_ddate,5,2),'-
   ',substr(b.princpl_overdue_starting_ddate,7,2)),
3  concat(substr(b.interst_overdue_starting_ddate,1,4),'-
   ',substr(b.interst_overdue_starting_ddate,5,2),'-
   ',substr(b.interst_overdue_starting_ddate,7,2)))
4  )+1
5
```

## 2、Hive 里面的DateDiff 函数

```
1  select datediff('2023-10-01', '2023-09-28');
```

对应Tchouse:

```

1 select date_part('day', '2023-10-01'::date - '2023-09-28'::date);
2 函数参数： 第一个是周期： day, week, month, year
3 第二个参数： endDate
4 第三个参数： startDate

```

## hive months\_between(endDate,startDate)

```

1 --计算两个日期相差的月份：months_between(endDate,startDate),日期格式为yyyy-mm-dd, 如果日期格
  式为yyyymmdd,需转化为yyyy-mm-dd, from_unixtime(unix_timestamp('yyyymmdd'),'yyyy-mm-dd');
2      ,round(months_between(concat(substring(due_date,1,4),'-
  ',substring(due_date,5,2),'-',substring(due_date,7,2)),
3      concat(substring(starting_date,1,4),'-',substring(starting_date,5,2),'-
  ',substring(starting_date,7,2))),0) as loan_length --贷款期限

```

转化为Tchouse计算两个时间的相差月份

```

1

```

## hive 出生年月份 判断生肖

```

1 select
2 SUBSTR('鼠牛虎兔龙蛇马羊猴鸡狗猪',CAST ( ( ( SUBSTR(birth_date ,1 ,4)-1900
3                                     ) % 12
4                                     )+1 AS INT ) ,1);

```

验证

```

mydb=# SUBSTR('鼠牛虎兔龙蛇马羊猴鸡狗猪',CAST ( ( ( SUBSTR('20231210' ,1 ,4)-1900
mydb=#                                     ) % 12
mydb=#                                     )+1 AS INT ) ,1);
 substr
-----
 兔
(1 row)

```

Tchouse 语法和hive一致

## Hive 里面decode的用法

```

1 SELECT DECODE(SUBSTR('鼠牛虎兔龙蛇马羊猴鸡狗猪',CAST ( ( ( SUBSTR('20231210',1,4)-1900
2                                ) % 12
3                                )+1 AS INT ),1), '鼠', '1', '牛'
4                                , '2', '虎', '3', '兔'
5                                , '4', '龙', '5', '蛇'
6                                , '6', '马', '7', '羊'
7                                , '8', '猴', '9', '鸡'
8                                , '10', '狗', '11', '猪'
9                                , '12') as zodiac_info;

```

tcchouse的用法一致，验证:

```

mydb=# SELECT DECODE(SUBSTR('鼠牛虎兔龙蛇马羊猴鸡狗猪',CAST ( ( ( SUBSTR('20231210',1,4)-1900
mydb(#                                ) % 12
mydb(#                                )+1 AS INT ),1), '鼠', '1', '牛'
mydb(#                                , '2', '虎', '3', '兔'
mydb(#                                , '4', '龙', '5', '蛇'
mydb(#                                , '6', '马', '7', '羊'
mydb(#                                , '8', '猴', '9', '鸡'
mydb(#                                , '10', '狗', '11', '猪'
mydb(#                                , '12') as zodiac_info;
zodiac_info
-----
4
(1 row)

```

根据身份证号码来判断性别， 年龄， 地域

```

1      SELECT
2          case left(id_card,2)
3              when "11" then "北京市"
4              when "12" then "天津市"
5              when "13" then "河北省"
6              when "14" then "山西省"
7              when "15" then "内蒙古自治区"
8              when "21" then "辽宁省"
9              when "22" then "吉林省"
10             when "23" then "黑龙江省"
11             when "31" then "上海市"
12             when "32" then "江苏省"
13             when "33" then "浙江省"
14             when "34" then "安徽省"
15             when "35" then "福建省"
16             when "36" then "江西省"
17             when "37" then "山东省"
18             when "41" then "河南省"
19             when "42" then "湖北省"
20             when "43" then "湖南省"
21             when "44" then "广东省"
22             when "45" then "广西壮族自治区"
23             when "46" then "海南省"
24             when "50" then "重庆市"
25             when "51" then "四川省"
26             when "52" then "贵州省"
27             when "53" then "云南省"
28             when "54" then "西藏自治区"
29             when "61" then "陕西省"
30             when "62" then "甘肃省"
31             when "63" then "青海省"
32             when "64" then "宁夏回族自治区"
33             when "65" then "新疆维吾尔自治区"
34             when "71" then "台湾省"
35             when "81" then "香港特别行政区"
36             when "82" then "澳门特别行政区"
37             else "未知"
38         end AS 地域 ,
39         year(curdate())-
40         if(length(id_card)=18,substring(id_card,7,4),if(length(id_card)=15,concat("19",substring(id_card,7,4)),
41         as 年龄,
42         case if(length(id_card)=18, cast(substring(id_card,17,1) as int)%2,
43             if(length(id_card)=15,cast(substring(id_card,15,1) as int)%2,3))
44             when 1 then "男"
45             when 0 then "女"
46             else "未知"

```

```
44     end AS 性别
45     FROM table_name
```

验证:

```
mydb=# select
mydb=# case if(length(a.id_card)=18, cast(substring(a.id_card,17,1) as int)%2,
mydb=# if(length(a.id_card)=15,cast(substring(a.id_card,15,1) as int)%2,3))
mydb=#   when 1 then '男'
mydb=#   when 0 then '女'
mydb=#   else '未知'
mydb=#   end AS 性别
mydb=#   from (
mydb=# select '622722199702121710' as id_card) a ;
  性别
-----
   男
(1 row)
```

## Hive 里面isdate 函数

```
1    ISDATE (expression, 'yyyyMMdd')
2
```

其中，expression表示要判断是否为日期类型的字符串或表达式。

该函数会返回一个布尔值，如果expression可以转换成日期类型，则返回1，否则返回0。

Tchouse 改写为:

```
1    create or replace function isdate(expression text, format text)
2    returns boolean
3    as
4    $$
5    declare result boolean;
6    begin
7    /**
8     * 功能说明：判断一个日期字符串是否是给定格式的日期字符串
9     * 参数：expression 是日期字符串  format 日期格式
10    *
11    */
12    result := false;
13    if length(expression) = length(format) and substr(expression, 5,1) = substr(format, 5,1)
14    then
15        result := true;
16    end if;
17    return result;
18    end;
19
20    $$
21    language plpgsql;
```



验证:

```
CREATE FUNCTION
mydb=# select isdate('2023-10-01', 'yyyymmdd');
isdate
-----
f
(1 row)

mydb=# select isdate('2023-10-01', 'yyyy-mm-dd');
isdate
-----
t
(1 row)

mydb=# select isdate('2023/10/01', 'yyyy-mm-dd');
isdate
-----
f
(1 row)
```

## Tchouse 里面 从身份证号码获取出生日期

其中需要处理15 18位身份证的特殊情况

```
1  create or replace function get_idcard_birth_date(id_card text)
2  returns text
3  as
4  $$
5  declare birth_date text;
6  begin
7  /**
8   * 功能说明： 从身份证中获取出生年月日 需要处理15位和18位身份证的特殊情况
9   *
10  *
11  * */
12  birth_date:=null;
13  if length(id_card) = 18 then
14    birth_date:=substring(id_card, 7,8);
15  elsif length(id_card) = 15 then
16    birth_date:=concat('19',substring(id_card, 7,6));
17  end if;
18  return birth_date;
19  end;
20
21  $$
22  language plpgsql;
```

```
CREATE FUNCTION
mydb=# create or replace function get_idcard_birth_date(id_card text)
mydb=# returns text
mydb=# as
mydb=# $$
mydb$# declare birth_date text;
mydb$# begin
mydb$# /**
mydb$# * 功能说明： 从身份证中获取出生年月日 需要处理15位和18位身份证的特殊情况
mydb$# *
mydb$# *
mydb$# * */
mydb$# birth_date:=null;
mydb$# if length(id_card) = 18 then
mydb$# birth_date:=substring(id_card, 7,8);
mydb$# elsif length(id_card) = 15 then
mydb$# birth_date:=concat('19',substring(id_card, 7,6));
mydb$# end if;
mydb$# return birth_date;
mydb$# end;
mydb$# $$
mydb=# language plpgsql;
CREATE FUNCTION
mydb=# select get_idcard_birth_date('622722199710231234')::date;
get_idcard_birth_date
-----
1997-10-23
(1 row)

mydb=# select get_idcard_birth_date('622722971023120')::date;
get_idcard_birth_date
-----
1997-10-23
(1 row)
```