# 最强最全面的大数据 SQL 面试题和答案

V1.0

本文档来自公众号：**五分钟学大数据**

**微信扫码直接关注**

# 目录

本套 SQL 题的答案是由许多小伙伴共同贡献的，1+1 的力量是远远大于 2 的，有不少题目都采用了**非常巧妙的解法**，也有不少题目**有多种解法**。本套大数据 SQL 题不仅题目丰富多样，答案更是精彩绝伦！

非常感谢**五分钟学大数据读者群**以下大佬的贡献：Mr.S、后会无期、我就是我、涛声依旧、徐明、蛋白、只争朝夕、魏明、十六画生、John.Xiong、南无、。、瓜皮、Camino、认真的小眼睛、热爱生活 Hello World！、无语梦醒、学不动了、life、执笔者、瓜皮、清风明月、焕溪沙、Dragon・、惟吾德馨 LY、长夜未央、欧哈呦、姜明松、乔一、小强、情深@骚明

因时间及水平有限，且避免不了因疏忽等情况导致答案出错，如您发现答案有错误或您有更优解，欢迎加我微信(yuan_more)告知，感激不尽！


注：以下参考答案都经过简单数据场景进行测试通过，但并未测试其他复杂情况。本文档的 SQL 主要使用 Hive SQL。

# 一、行列转换

**描述**：表中记录了各年份各部门的平均绩效考核成绩。

表名：t1

表结构：

```
a -- 年份
b -- 部门
c -- 绩效得分
```

**表内容**：

```
 a    b   c
2014  B   9
2015  A   8
2014  A   10
2015  B   7
```

## 问题一：多行转多列

**问题描述**：将上述表内容转为如下输出结果所示：

```
 a   col_A col_B
2014  10    9
2015  8     7
```

**参考答案**：

```
select
    a,
    max(case when b="A" then c end) col_A,
    max(case when b="B" then c end) col_B
from t1
group by a;
```

## 问题二：如何将结果转成源表？（多列转多行）

**问题描述**：将问题一的结果转成源表，问题一结果表名为 t1_2。

**参考答案**：

```
select
    a,
    b,
    c
```

```
from (
    select a,"A" as b,col_a as c from t1_2
    union all
    select a,"B" as b,col_b as c from t1_2
)tmp;
```

## 问题三：同一部门会有多个绩效，求多行转多列结果

**问题描述**：2014 年公司组织架构调整，导致部门出现多个绩效，业务及人员不同，无法合并算绩效，源表内容如下：

```
2014  B  9
2015  A  8
2014  A  10
2015  B  7
2014  B  6
```

**输出结果如下所示**：

```
 a    col_A  col_B
2014   10    6,9
2015   8     7
```

**参考答案**：

```
select
    a,
    max(case when b="A" then c end) col_A,
    max(case when b="B" then c end) col_B
from (
    select
        a,
        b,
        concat_ws(",",collect_set(cast(c as string))) as c
    from t1
    group by a,b
)tmp
group by a;
```

## 二、排名中取他值

表名：t2

表字段及内容：

```
a    b   c
2014 A   3
2014 B   1
2014 C   2
2015 A   4
2015 D   3
```

### 问题一：按 a 分组取 b 字段最小时对应的 c 字段

输出结果如下所示：

```
a   min_c
2014  3
2015  4
```

参考答案：

```
select
  a,
  c as min_c
from
(
      select
        a,
        b,
        c,
```

```
        row_number() over(partition by a order by b) as rn
      from t2
)a
where rn = 1;
```

## 问题二：按 a 分组取 b 字段排第二时对应的 c 字段

输出结果如下所示：

```
 a  second_c
2014  1
2015  3
```

参考答案：

```
select
  a,
  c as second_c
from
(
      select
        a,
        b,
        c,
        row_number() over(partition by a order by b) as rn
      from t2
)a
where rn = 2;
```

## 问题三：按 a 分组取 b 字段最小和最大时对应的 c 字段

输出结果如下所示：

```
a    min_c  max_c
2014  3      2
2015  4      3
```

参考答案：

```
select
  a,
  min(if(asc_rn = 1, c, null)) as min_c,
  max(if(desc_rn = 1, c, null)) as max_c
from
(
```

```
      select
        a,
        b,
        c,
        row_number() over(partition by a order by b) as asc_rn,
        row_number() over(partition by a order by b desc) as desc_rn
      from t2
)a
where asc_rn = 1 or desc_rn = 1
group by a;
```

## 问题四：按 a 分组取 b 字段第二小和第二大时对应的 c 字段

输出结果如下所示：

```
a    min_c  max_c
2014  1      1
2015  3      4
```

参考答案：

```
select
    ret.a
    ,max(case when ret.rn_min = 2 then ret.c else null end) as min_c
    ,max(case when ret.rn_max = 2 then ret.c else null end) as max_c
from (
    select
        *
        ,row_number() over(partition by t2.a order by t2.b) as rn_min
        ,row_number() over(partition by t2.a order by t2.b desc) as rn_max
    from t2
) as ret
where ret.rn_min = 2
or ret.rn_max = 2
group by ret.a;
```

## 问题五：按 a 分组取 b 字段前两小和前两大时对应的 c 字段

注意：需保持 b 字段最小、最大排首位

输出结果如下所示：

```
a    min_c  max_c
2014  3,1     2,1
2015  4,3     3,4
```

参考答案：

```
select
  tmp1.a as a,
  min_c,
  max_c
from
(
  select
    a,
    concat_ws(',', collect_list(c)) as min_c
  from
    (
     select
       a,
       b,
       c,
       row_number() over(partition by a order by b) as asc_rn
     from t2
     )a
    where asc_rn <= 2
    group by a
)tmp1
join
(
  select
    a,
    concat_ws(',', collect_list(c)) as max_c
  from
    (
     select
        a,
        b,
        c,
        row_number() over(partition by a order by b desc) as desc_rn
     from t2
    )a
    where desc_rn <= 2
    group by a
)tmp2
on tmp1.a = tmp2.a;
```

## 三、累计求值

表名：t3

表字段及内容：

```
a    b   c
2014  A   3
2014  B   1
2014  C   2
2015  A   4
2015  D   3
```

### 问题一：按 a 分组按 b 字段排序，对 c 累计求和

输出结果如下所示：

```
a    b   sum_c
2014  A   3
2014  B   4
2014  C   6
2015  A   4
2015  D   7
```

参考答案：

```
select
  a,
  b,
  c,
  sum(c) over(partition by a order by b) as sum_c
from t3;
```

### 问题二：按 a 分组按 b 字段排序，对 c 取累计平均值

输出结果如下所示：

```
a    b   avg_c
2014  A   3
2014  B   2
2014  C   2
2015  A   4
2015  D   3.5
```

参考答案：

```
select
  a,
  b,
  c,
  avg(c) over(partition by a order by b) as avg_c
from t3;
```

## 问题三：按 a 分组按 b 字段排序，对 b 取累计排名比例

输出结果如下所示：

```
a    b   ratio_c
2014  A   0.33
2014  B   0.67
2014  C   1.00
2015  A   0.50
2015  D   1.00
```

参考答案：

```
select
  a,
  b,
  c,
  round(row_number() over(partition by a order by b) / (count(c) over(partition by
a)),2) as ratio_c
from t3
order by a,b;
```

## 问题四：按 a 分组按 b 字段排序，对 b 取累计求和比例

输出结果如下所示：

```
a    b   ratio_c
2014  A   0.50
2014  B   0.67
2014  C   1.00
2015  A   0.57
2015  D   1.00
```

参考答案：

```
select
  a,
  b,
```

```
  c,
  round(sum(c) over(partition by a order by b) / (sum(c) over(partition by a)),2) a
s ratio_c
from t3
order by a,b;
```

## 四、窗口大小控制

表名：t4

表字段及内容：

```
a    b   c
2014  A   3
2014  B   1
2014  C   2
2015  A   4
2015  D   3
```

### 问题一：按 a 分组按 b 字段排序，对 c 取前后各一行的和

输出结果如下所示：

```
a    b   sum_c
2014  A   1
2014  B   5
2014  C   1
2015  A   3
2015  D   4
```

参考答案：

```
select
  a,
  b,
  lag(c,1,0) over(partition by a order by b)+lead(c,1,0) over(partition by a order
by b) as sum_c
from t4;
```

### 问题二：按 a 分组按 b 字段排序，对 c 取平均值

问题描述：前一行与当前行的均值！

输出结果如下所示：

```
a    b   avg_c
2014  A   3
2014  B   2
2014  C   1.5
2015  A   4
2015  D   3.5
```

参考答案：

```
select
  a,
  b,
  case when lag_c is null then c
  else (c+lag_c)/2 end as avg_c
from
 (
 select
   a,
   b,
   c,
   lag(c,1) over(partition by a order by b) as lag_c
  from t4
 )temp;
```

## 五、产生连续数值

输出结果如下所示：

```
1
2
3
4
5
...
100
```

参考答案：

不借助其他任何外表，实现产生连续数值

此处给出两种解法，其一：

```
select
id_start+pos as id
from(
    select
    1 as id_start,
    1000000 as id_end
) m  lateral view posexplode(split(space(id_end-id_start), '')) t as pos, val
```

其二：

```
select
  row_number() over() as id
from
  (select split(space(99), ' ') as x) t
lateral view
explode(x) ex;
```

那如何产生 1 至 1000000 连续数值？

参考答案：

```
select
  row_number() over() as id
from
  (select split(space(999999), ' ') as x) t
lateral view
explode(x) ex;
```

## 六、数据扩充与收缩

表名：t6

表字段及内容：

```
a
3
2
4
```

### 问题一：数据扩充

输出结果如下所示：

```
a     b
3   3、2、1
2   2、1
4   4、3、2、1
```

参考答案：

```
select
  t.a,
  concat_ws('、',collect_set(cast(t.rn as string))) as b
from
(
  select
```

```
    t6.a,
    b.rn
  from t6
  left join
  (
   select
     row_number() over() as rn
   from
   (select split(space(5), ' ') as x) t -- space(5)可根据 t6 表的最大值灵活调整
   lateral view
   explode(x) pe
  ) b
  on 1 = 1
  where t6.a >= b.rn
  order by t6.a, b.rn desc
) t
group by  t.a;
```

## 问题二：数据扩充，排除偶数

输出结果如下所示：

```
a     b
3   3、1
2   1
4   3、1
```

参考答案：

```
select
  t.a,
  concat_ws('、',collect_set(cast(t.rn as string))) as b
from
(
  select
    t6.a,
    b.rn
  from t6
  left join
  (
   select
     row_number() over() as rn
   from
   (select split(space(5), ' ') as x) t
```

```
   lateral view
   explode(x) pe
  ) b
  on 1 = 1
  where t6.a >= b.rn and b.rn % 2 = 1
  order by t6.a, b.rn desc
) t
group by  t.a;
```

## 问题三：如何处理字符串累计拼接

问题描述：将小于等于 a 字段的值聚合拼接起来

输出结果如下所示：

```
a     b
3     2、3
2     2
4     2、3、4
```

参考答案：

```
select
  t.a,
  concat_ws('、',collect_set(cast(t.a1 as string))) as b
from
(
  select
    t6.a,
    b.a1
  from t6
  left join
  (
   select  a as a1
   from t6
  ) b
  on 1 = 1
  where t6.a >= b.a1
  order by t6.a, b.a1
) t
group by  t.a;
```

## 问题四：如果 a 字段有重复，如何实现字符串累计拼接

输出结果如下所示：

```
a     b
2     2
3     2、3
3     2、3、3
4     2、3、3、4
```

参考答案：

```
select
  a,
  b
from
(
 select
   t.a,
   t.rn,
   concat_ws('、',collect_list(cast(t.a1 as string))) as b
 from
  (
    select
     a.a,
     a.rn,
     b.a1
    from
    (
     select
       a,
       row_number() over(order by a ) as rn
     from t6
    ) a
    left join
    (
     select  a as a1,
     row_number() over(order by a ) as rn
     from t6
    ) b
    on 1 = 1
    where a.a >= b.a1 and a.rn >= b.rn
    order by a.a, b.a1
  ) t
  group by  t.a,t.rn
  order by t.a,t.rn
) tt;
```

## 问题五：数据展开

**问题描述**：如何将字符串"1-5,16,11-13,9"扩展成"1,2,3,4,5,16,11,12,13,9"？注意顺序不变。

**参考答案**：

```
select
  concat_ws(',',collect_list(cast(rn as string)))
from
(
  select
   a.rn,
   b.num,
   b.pos
  from
   (
    select
     row_number() over() as rn
    from (select split(space(20), ' ') as x) t -- space(20)可灵活调整
    lateral view
    explode(x) pe
   ) a lateral view outer
   posexplode(split('1-5,16,11-13,9', ',')) b as pos, num
   where a.rn between cast(split(num, '-')[0] as int) and cast(split(num, '-')[1] a
s int) or a.rn = num
   order by pos, rn
) t;
```



## 七、合并与拆分

表名：t7

表字段及内容：

```
a    b
2014  A
2014  B
2015  B
2015  D
```

## 问题一：合并

输出结果如下所示：

```
2014  A、B
2015  B、D
```

参考答案：

```
select
  a,
  concat_ws('、', collect_set(t.b)) b
from t7
group by a;
```

## 问题二：拆分

问题描述：将分组合并的结果拆分出来

参考答案：

```
select
  t.a,
  d
from
(
 select
  a,
  concat_ws('、', collect_set(t7.b)) b
 from t7
 group by a
)t
lateral view
explode(split(t.b, '、')) table_tmp as d;
```

## 八、模拟循环操作

表名：t8

表字段及内容：

```
a
1011
0101
```

**问题一：如何将字符’1’的位置提取出来**

输出结果如下所示：

```
1,3,4
2,4
```

参考答案：

```
select
    a,
    concat_ws(",",collect_list(cast(index as string))) as res
from (
    select
        a,
        index+1 as index,
        chr
    from (
        select
            a,
            concat_ws(",",substr(a,1,1),substr(a,2,1),substr(a,3,1),substr(a,-1)) s
tr
        from t8
    ) tmp1
    lateral view posexplode(split(str,",")) t as index,chr
    where chr = "1"
) tmp2
group by a;
```

## 九、不使用 distinct 或 group by 去重

表名：t9

表字段及内容：

```
a     b     c     d
2014  2016  2014   A
2014  2015  2015   B
```

## 问题一：不使用 distinct 或 group by 去重

输出结果如下所示：

```
2014  A
2016  A
2014  B
2015  B
```

参考答案：

```
select
  t2.year
  ,t2.num
from
 (
  select
    *
    ,row_number() over (partition by t1.year,t1.num) as rank_1
  from
  (
    select
      a as year,
      d as num
    from t9
    union all
    select
      b as year,
      d as num
    from t9
    union all
    select
      c as year,
      d as num
    from t9
   )t1
)t2
where rank_1=1
order by num;
```

## 十、容器--反转内容

表名：t10

表字段及内容：

```
a
AB,CA,BAD
BD,EA
```

## 问题一：反转逗号分隔的数据：改变顺序，内容不变

输出结果如下所示：

```
BAD,CA,AB
EA,BD
```

参考答案：

```
select
  a,
  concat_ws(",",collect_list(reverse(str)))
from
(
  select
    a,
    str
  from t10
  lateral view explode(split(reverse(a),",")) t as str
) tmp1
group by a;
```

## 问题二：反转逗号分隔的数据：改变内容，顺序不变

输出结果如下所示：

```
BA,AC,DAB
DB,AE
```

参考答案：

```
select
  a,
  concat_ws(",",collect_list(reverse(str)))
from
(
  select
     a,
     str
  from t10
  lateral view explode(split(a,",")) t as str
```

```
) tmp1
group by a;
```

## 十一、多容器--成对提取数据

表名：t11

表字段及内容：

```
a       b
A/B     1/3
B/C/D   4/5/2
```

### 问题一：成对提取数据，字段一一对应

输出结果如下所示：

```
a       b
A       1
B       3
B       4
C       5
D       2
```

参考答案：

```
select
  a_inx,
  b_inx
from
(
  select
     a,
     b,
     a_id,
     a_inx,
     b_id,
     b_inx
  from t11
  lateral view posexplode(split(a,'/')) t as a_id,a_inx
  lateral view posexplode(split(b,'/')) t as b_id,b_inx
) tmp
where a_id=b_id;
```

## 十二、多容器--转多行

表名：t12

表字段及内容：

```
a        b       c
001      A/B     1/3/5
002      B/C/D   4/5
```

## 问题一：转多行

输出结果如下所示：

```
a        d        e
001      type_b     A
001      type_b     B
001      type_c     1
001      type_c     3
001      type_c     5
002      type_b     B
002      type_b     C
002      type_b     D
002      type_c     4
002      type_c     5
```

参考答案：

```
select
  a,
  d,
  e
from
(
  select
    a,
    "type_b" as d,
    str as e
  from t12
  lateral view explode(split(b,"/")) t as str
  union all
  select
    a,
    "type_c" as d,
    str as e
  from t12
  lateral view explode(split(c,"/")) t as str
```

```
) tmp
order by a,d;
```

## 十三、抽象分组--断点排序

表名：t13

表字段及内容：

```
a    b
2014  1
2015  1
2016  1
2017  0
2018  0
2019  -1
2020  -1
2021  -1
2022  1
2023  1
```

### 问题一：断点排序

输出结果如下所示：

```
a    b    c
2014  1    1
2015  1    2
2016  1    3
2017  0    1
2018  0    2
2019  -1   1
2020  -1   2
2021  -1   3
2022  1    1
2023  1    2
```

参考答案：

```
select
  a,
  b,
  row_number() over( partition by b,repair_a order by a asc) as c--按照 b 列和[b 的组首]
分组，排序
from
(
```

```
select
  a,
  b,
  a-b_rn as repair_a--根据b列值出现的次序,修复a列值为b首次出现的a列值,称为b的[组首]
from
(
 select
   a,
   b,
   row_number() over( partition by b order by  a  asc ) as b_rn--按b列分组,按a列排序,得到b列各值出现的次序
  from t13
)tmp1
)tmp2--注意，如果不同的b列值，可能出现同样的组首值，但组首值需要和a列值 一并参与分组，故并不影响排序。
order by a asc;
```

## 十四、业务逻辑的分类与抽象--时效

日期表：d_date

表字段及内容：

| date_id | is_work |
|---|---|
| 2017-04-13 | 1 |
| 2017-04-14 | 1 |
| 2017-04-15 | 0 |
| 2017-04-16 | 0 |
| 2017-04-17 | 1 |

工 作 日 ：周一至周五 09:30-18:30

客户申请表：t14

表字段及内容：

| a | b | c |
|---|---|---|
| 1 | 申请 | 2017-04-14 18:03:00 |
| 1 | 通过 | 2017-04-17 09:43:00 |
| 2 | 申请 | 2017-04-13 17:02:00 |
| 2 | 通过 | 2017-04-15 09:42:00 |

### 问题一：计算上表中从申请到通过占用的工作时长

输出结果如下所示：

```
a         d
1        0.67h
2       10.67h
```

参考答案：

```
select
    a,
    round(sum(diff)/3600,2) as d
from (
    select
        a,
        apply_time,
        pass_time,
        dates,
        rn,
        ct,
        is_work,
        case when is_work=1 and rn=1 then unix_timestamp(concat(dates,' 18:30:00'),
'yyyy-MM-dd HH:mm:ss')-unix_timestamp(apply_time,'yyyy-MM-dd HH:mm:ss')
            when is_work=0 then 0
            when is_work=1 and rn=ct then unix_timestamp(pass_time,'yyyy-MM-dd HH:m
m:ss')-unix_timestamp(concat(dates,' 09:30:00'),'yyyy-MM-dd HH:mm:ss')
            when is_work=1 and rn!=ct then 9*3600
        end diff
    from (
        select
            a,
            apply_time,
            pass_time,
            time_diff,
            day_diff,
            rn,
            ct,
            date_add(start,rn-1) dates
        from (
            select
                a,
                apply_time,
                pass_time,
                time_diff,
                day_diff,
                strs,
                start,
                row_number() over(partition by a) as rn,
```

```
                count(*) over(partition by a) as ct
            from (
                select
                    a,
                    apply_time,
                    pass_time,
                    time_diff,
                    day_diff,
                    substr(repeat(concat(substr(apply_time,1,10),','),day_diff+1),1,
11*(day_diff+1)-1) strs
                from (
                    select
                        a,
                        apply_time,
                        pass_time,
                        unix_timestamp(pass_time,'yyyy-MM-dd HH:mm:ss')-unix_timest
amp(apply_time,'yyyy-MM-dd HH:mm:ss') time_diff,
                        datediff(substr(pass_time,1,10),substr(apply_time,1,10)) da
y_diff
                    from (
                        select
                            a,
                            max(case when b='申请' then c end) apply_time,
                            max(case when b='通过' then c end) pass_time
                        from t14
                        group by a
                    ) tmp1
                ) tmp2
            ) tmp3
            lateral view explode(split(strs,",")) t as start
        ) tmp4
    ) tmp5
    join d_date
    on tmp5.dates = d_date.date_id
) tmp6
group by a;
```

## 十五、时间序列--进度及剩余

表名：t15

表字段及内容：

```
date_id      is_work
2017-07-30      0
2017-07-31      1
```

```
2017-08-01      1
2017-08-02      1
2017-08-03      1
2017-08-04      1
2017-08-05      0
2017-08-06      0
2017-08-07      1
```

## 问题一：求每天的累计周工作日，剩余周工作日

输出结果如下所示：

```
date_id      week_to_work   week_left_work
2017-07-31      1              4
2017-08-01      2              3
2017-08-02      3              2
2017-08-03      4              1
2017-08-04      5              0
2017-08-05      5              0
2017-08-06      5              0
```

参考答案：

此处给出两种解法，其一：

```
select
 date_id
,case date_format(date_id,'u')
    when 1 then 1
    when 2 then 2
    when 3 then 3
    when 4 then 4
    when 5 then 5
    when 6 then 5
    when 7 then 5
 end as week_to_work
,case date_format(date_id,'u')
    when 1 then 4
    when 2 then 3
    when 3 then 2
    when 4 then 1
    when 5 then 0
    when 6 then 0
    when 7 then 0
```

```
 end as week_to_work
from t15
```

其二：

```
select
date_id,
week_to_work,
week_sum_work-week_to_work as week_left_work
from(
    select
    date_id,
    sum(is_work) over(partition by year,week order by date_id) as week_to_work,
    sum(is_work) over(partition by year,week) as week_sum_work
    from(
        select
        date_id,
        is_work,
        year(date_id) as year,
        weekofyear(date_id) as week
        from t15
    ) ta
) tb order by date_id;
```



## 十六、时间序列--构造日期

### 问题一：直接使用 SQL 实现一张日期维度表，包含以下字段：

| | | |
|---|---|---|
| date | string | 日期 |
| d_week | string | 年内第几周 |
| weeks | int | 周几 |
| w_start | string | 周开始日 |
| w_end | string | 周结束日 |
| d_month | int | 第几月 |

| | | |
|---|---|---|
| m_start | string | 月开始日 |
| m_end | string | 月结束日 |
| d_quarter | int | 第几季 |
| q_start | string | 季开始日 |
| q_end | string | 季结束日 |
| d_year | int | 年份 |
| y_start | string | 年开始日 |
| y_end | string | 年结束日 |

## 参考答案：

```
drop table if exists dim_date;
create table if not exists dim_date(
    `date` string comment '日期',
    d_week string comment '年内第几周',
    weeks string comment '周几',
    w_start string comment '周开始日',
    w_end string comment '周结束日',
    d_month string comment '第几月',
    m_start string comment '月开始日',
    m_end string comment '月结束日',
    d_quarter int comment '第几季',
    q_start string comment '季开始日',
    q_end string comment '季结束日',
    d_year int comment '年份',
    y_start string comment '年开始日',
    y_end string comment '年结束日'
);
--自然月：指每月的 1 号到那个月的月底，它是按照阳历来计算的。就是从每月 1 号到月底，不管这个月有 30 天，31 天，29 天或者 28 天，都算是一个自然月。

insert overwrite table dim_date
select `date`
     , d_week --年内第几周
     , case weekid
           when 0 then '周日'
           when 1 then '周一'
           when 2 then '周二'
           when 3 then '周三'
           when 4 then '周四'
           when 5 then '周五'
           when 6 then '周六'
    end  as weeks -- 周
     , date_add(next_day(`date`,'MO'),-7) as w_start --周一
     , date_add(next_day(`date`,'MO'),-1) as w_end   -- 周日_end
```

```
    -- 月份日期
    , concat('第', monthid, '月')  as d_month
    , m_start
    , m_end

    -- 季节
    , quarterid as d_quart
    , concat(d_year, '-', substr(concat('0', (quarterid - 1) * 3 + 1), -2), '-01')
 as q_start --季开始日
    , date_sub(concat(d_year, '-', substr(concat('0', (quarterid) * 3 + 1), -2), '
-01'), 1) as q_end   --季结束日
    -- 年
    , d_year
    , y_start
    , y_end


from (
         select `date`
              , pmod(datediff(`date`, '2012-01-01'), 7)                 as weekid
   --获取周几
              , cast(substr(`date`, 6, 2) as int)                       as monthid
   --获取月份
              , case
                    when cast(substr(`date`, 6, 2) as int) <= 3 then 1
                    when cast(substr(`date`, 6, 2) as int) <= 6 then 2
                    when cast(substr(`date`, 6, 2) as int) <= 9 then 3
                    when cast(substr(`date`, 6, 2) as int) <= 12 then 4
                end                                                     as quarterid
 --获取季节 可以直接使用 quarter(`date`)
              , substr(`date`, 1, 4)                                    as d_year
   -- 获取年份
              , trunc(`date`, 'YYYY')                                   as y_start
   --年开始日
              , date_sub(trunc(add_months(`date`, 12), 'YYYY'), 1) as y_end     --年
结束日
              , date_sub(`date`, dayofmonth(`date`) - 1)                as m_start
   --当月第一天
              , last_day(date_sub(`date`, dayofmonth(`date`) - 1))          m_end
   --当月最后一天
              , weekofyear(`date`)                                      as d_week
   --年内第几周
         from (
                  -- '2021-04-01'是开始日期, '2022-03-31'是截止日期
```

```
                select date_add('2021-04-01', t0.pos) as `date`
                from (
                         select posexplode(
                                        split(
                                                repeat('o', datediff(
                                                        from_unixtime(unix_timest
amp('2022-03-31', 'yyyy-mm-dd'),
                                                                      'yyyy-mm-dd
'),
                                                        '2021-04-01')), 'o'
                                            )
                                    )
                     ) t0
             ) t1
    ) t2;
```

## 十七、时间序列--构造累积日期

表名：t17

表字段及内容：

```
date_id
2017-08-01
2017-08-02
2017-08-03
```

### 问题一：每一日期，都扩展成月初至当天

输出结果如下所示：

```
date_id    date_to_day
2017-08-01  2017-08-01
2017-08-02  2017-08-01
2017-08-02  2017-08-02
2017-08-03  2017-08-01
2017-08-03  2017-08-02
2017-08-03  2017-08-03
```

这种累积相关的表，常做桥接表。

参考答案：

```
select
  date_id,
  date_add(date_start_id,pos) as date_to_day
from
```

```
(
  select
    date_id,
    date_sub(date_id,dayofmonth(date_id)-1) as date_start_id
  from t17
) m  lateral view
posexplode(split(space(datediff(from_unixtime(unix_timestamp(date_id,'yyyy-MM-dd')),
from_unixtime(unix_timestamp(date_start_id,'yyyy-MM-dd')))), '')) t as pos, val;
```

## 十八、时间序列--构造连续日期

表名：t18

表字段及内容：

```
a             b          c
101         2018-01-01     10
101         2018-01-03     20
101         2018-01-06     40
102         2018-01-02     20
102         2018-01-04     30
102         2018-01-07     60
```

### 问题一：构造连续日期

问题描述：将表中数据的 b 字段扩充至范围[2018-01-01, 2018-01-07]，并累积对 c 求和。

b 字段的值是较稀疏的。

输出结果如下所示：

```
a             b          c      d
101         2018-01-01     10     10
101         2018-01-02      0     10
101         2018-01-03     20     30
101         2018-01-04      0     30
101         2018-01-05      0     30
101         2018-01-06     40     70
101         2018-01-07      0     70
102         2018-01-01      0      0
102         2018-01-02     20     20
102         2018-01-03      0     20
102         2018-01-04     30     50
102         2018-01-05      0     50
```

```
102         2018-01-06      0     50
102         2018-01-07     60    110
```

参考答案：

```
select
  a,
  b,
  c,
  sum(c) over(partition by a order by b) as d
from
(
  select
  t1.a,
  t1.b,
  case
    when t18.b is not null then t18.c
    else 0
  end as c
  from
  (
    select
    a,
    date_add(s,pos) as b
    from
    (
      select
        a,
       '2018-01-01' as s,
       '2018-01-07' as r
      from (select a from t18 group by a) ta
    ) m  lateral view
      posexplode(split(space(datediff(from_unixtime(unix_timestamp(r,'yyyy-MM-dd')),
from_unixtime(unix_timestamp(s,'yyyy-MM-dd')))), '')) t as pos, val
  ) t1
    left join t18
    on  t1.a = t18.a and t1.b = t18.b
) ts;
```

## 十九、时间序列--取多个字段最新的值

表名：t19

表字段及内容：

```
date_id   a   b    c
2014     AB  12    bc
```

```
2015          23
2016                d
2017      BC
```

## 问题一：如何一并取出最新日期

输出结果如下所示：

```
date_a    a     date_b     b     date_c    c
2017      BC    2015       23    2016      d
```

参考答案：

此处给出三种解法，其一：

```
SELECT  max(CASE WHEN rn_a = 1 THEN date_id else 0 END) AS date_a
        ,max(CASE WHEN rn_a = 1 THEN a else null END) AS a
        ,max(CASE WHEN rn_b = 1 THEN date_id else 0 END) AS date_b
        ,max(CASE WHEN rn_b = 1 THEN b else NULL  END) AS b
        ,max(CASE WHEN rn_c = 1 THEN date_id  else 0 END) AS date_c
        ,max(CASE WHEN rn_c = 1 THEN c else null END) AS c
FROM    (
            SELECT  date_id
                    ,a
                    ,b
                    ,c
                    --对每列上不为 null 的值  的 日期 进行排序
                    ,row_number()OVER( PARTITION BY 1 ORDER BY CASE WHEN a IS NULL
THEN 0 ELSE date_id END DESC) AS rn_a
                    ,row_number()OVER(PARTITION BY 1 ORDER BY CASE WHEN b IS NULL T
HEN 0 ELSE date_id END DESC) AS rn_b
                    ,row_number()OVER(PARTITION BY 1 ORDER BY CASE WHEN c IS NULL T
HEN 0 ELSE date_id END DESC) AS rn_c
            FROM    t19
        ) t
WHERE   t.rn_a = 1
OR      t.rn_b = 1
OR      t.rn_c = 1;
```

其二：

```
SELECT
   a.date_id
  ,a.a
  ,b.date_id
  ,b.b
```

```
  ,c.date_id
  ,c.c
FROM
(
   SELECT
      t.date_id,
      t.a
   FROM
   (
     SELECT
       t.date_id
       ,t.a
       ,t.b
       ,t.c
     FROM t19 t INNER JOIN    t19 t1 ON t.date_id = t1.date_id AND t.a IS NOT NULL
   ) t
   ORDER BY t.date_id DESC
   LIMIT 1
) a
LEFT JOIN
(
  SELECT
    t.date_id
    ,t.b
  FROM
  (
    SELECT
      t.date_id
      ,t.b
    FROM t19 t INNER JOIN t19 t1 ON t.date_id = t1.date_id AND t.b IS NOT NULL
  ) t
  ORDER BY t.date_id DESC
  LIMIT 1
) b ON 1 = 1
LEFT JOIN
(
  SELECT
    t.date_id
    ,t.c
  FROM
  (
    SELECT
      t.date_id
      ,t.c
```

```
    FROM t19 t INNER JOIN t19 t1 ON t.date_id = t1.date_id AND t.c IS NOT NULL
  ) t
  ORDER BY t.date_id DESC
  LIMIT   1
) c
ON 1 = 1;
```

其三：

```
select
  *
from
(
  select t1.date_id as date_a,t1.a from (select t1.date_id,t1.a  from t19 t1 where
t1.a is not null) t1
  inner join (select max(t1.date_id) as date_id   from t19 t1 where t1.a is not nul
l) t2
  on t1.date_id=t2.date_id
) t1
cross join
(
  select t1.date_b,t1.b from (select t1.date_id as date_b,t1.b  from t19 t1 where t
1.b is not null) t1
  inner join (select max(t1.date_id) as date_id   from t19 t1 where t1.b is not nul
l)t2
  on t1.date_b=t2.date_id
) t2
cross join
(
  select t1.date_c,t1.c from (select t1.date_id as date_c,t1.c  from t19 t1 where t
1.c is not null) t1
  inner join (select max(t1.date_id) as date_id   from t19 t1 where t1.c is not nul
l)t2
  on t1.date_c=t2.date_id
) t3;
```

## 二十、时间序列--补全数据

表名：t20

表字段及内容：

```
date_id   a    b    c
2014      AB   12   bc
2015           23
2016                d
2017      BC
```

### 问题一：如何使用最新数据补全表格

输出结果如下所示：

```
date_id   a    b    c
2014      AB   12   bc
2015      AB   23   bc
2016      AB   23   d
2017      BC   23   d
```

参考答案：

```
select
  date_id,
  first_value(a) over(partition by aa order by date_id) as a,
  first_value(b) over(partition by bb order by date_id) as b,
  first_value(c) over(partition by cc order by date_id) as c
from
(
  select
    date_id,
    a,
    b,
    c,
    count(a) over(order by date_id) as aa,
    count(b) over(order by date_id) as bb,
    count(c) over(order by date_id) as cc
  from t20
)tmp1;
```

## 二十一、时间序列--取最新完成状态的前一个状态

表名：t21

表字段及内容：

```
date_id   a    b
2014      1    A
2015      1    B
2016      1    A
2017      1    B
2013      2    A
2014      2    B
2015      2    A
2014      3    A
2015      3    A
2016      3    B
2017      3    A
```

上表中 B 为完成状态。

## 问题一：取最新完成状态的前一个状态

输出结果如下所示：

```
date_id  a    b
2016     1    A
2013     2    A
2015     3    A
```

参考答案：

此处给出两种解法，其一：

```
select
    t21.date_id,
    t21.a,
    t21.b
from
    (
        select
            max(date_id) date_id,
            a
        from
            t21
        where
            b = 'B'
        group by
            a
```

```
    ) t1
    inner join t21 on t1.date_id -1 = t21.date_id
and t1.a = t21.a;
```

其二：

```
select
  next_date_id as date_id
  ,a
  ,next_b as b
from(
  select
    *,min(nk) over(partition by a,b) as minb
  from(
    select
      *,row_number() over(partition by a order by date_id desc) nk
      ,lead(date_id) over(partition by a order by date_id desc) next_date_id
      ,lead(b) over(partition by a order by date_id desc) next_b
    from(
      select * from t21
    ) t
  ) t
) t
where minb = nk and b = 'B';
```

## 问题二：如何将完成状态的过程合并

输出结果如下所示：

```
a   b_merge
1   A、B、A、B
2   A、B
3   A、A、B
```

参考答案：

```
select
  a
  ,collect_list(b) as b
from(
  select
    *
    ,min(if(b = 'B',nk,null)) over(partition by a) as minb
  from(
    select
      *,row_number() over(partition by a order by date_id desc) nk
```

```
    from(
      select * from t21
    ) t
  ) t
) t
where nk >= minb
group by a;
```

## 二十二、非等值连接--范围匹配

表 f 是事实表，表 d 是匹配表，在 hive 中如何将匹配表中的值关联到事实表中？表 d 相当于拉链过的变化维，但日期范围可能是不全的。

表 f：

```
date_id  p_id
 2017    C
 2018    B
 2019    A
 2013    C
```

表 d：

```
d_start    d_end    p_id    p_value
 2016      2018     A       1
 2016      2018     B       2
 2008      2009     C       4
 2010      2015     C       3
```

### 问题一：范围匹配

输出结果如下所示：

```
date_id  p_id   p_value
 2017    C      null
 2018    B      2
 2019    A      null
 2013    C      3
```

**参考答案：

此处给出两种解法，其一：

```
select
  f.date_id,
  f.p_id,
  A.p_value
from f
```

```
left join
(
  select
    date_id,
    p_id,
    p_value
  from
  (
    select
      f.date_id,
      f.p_id,
      d.p_value
    from f
    left join d on f.p_id = d.p_id
    where f.date_id >= d.d_start and f.date_id <= d.d_end
  )A
)A
ON f.date_id = A.date_id;
```

其二：

```
select
    date_id,
    p_id,
    flag as p_value
from (
    select
        f.date_id,
        f.p_id,
        d.d_start,
        d.d_end,
        d.p_value,
        if(f.date_id between d.d_start and d.d_end,d.p_value,null) flag,
        max(d.d_end) over(partition by date_id) max_end
    from f
    left join d
    on f.p_id = d.p_id
) tmp
where d_end = max_end;
```

## 二十三、非等值连接--最近匹配

表 t23_1 和表 t23_2 通过 a 和 b 关联时，有相等的取相等的值匹配，不相等时每一个 a 的值在 b 中找差值最小的来匹配。

t23_1 和 t23_2 为两个班的成绩单，t23_1 班的每个学生成绩在 t23_2 班中找出成绩最接近的成绩。

表 t23_1：a 中无重复值

```
a
1
2
4
5
8
10
```

表 t23_2：b 中无重复值

```
b
2
3
7
11
13
```

## 问题一：单向最近匹配

输出结果如下所示：

注意：b 的值可能会被丢弃

```
a    b
1    2
2    2
4    3
5    3
5    7
8    7
10   11
```

参考答案：

```
select
  *
from
(
  select
    ttt1.a,
    ttt1.b
  from
```

```
  (
    select
      tt1.a,
      t23_2.b,
      dense_rank() over(partition by tt1.a order by abs(tt1.a-t23_2.b)) as dr
    from
    (
      select
        t23_1.a
      from t23_1
      left join t23_2 on t23_1.a=t23_2.b
      where t23_2.b is null
    ) tt1
    cross join t23_2
  ) ttt1
  where ttt1.dr=1
  union all
  select
    t23_1.a,
    t23_2.b
  from t23_1
  inner join t23_2 on t23_1.a=t23_2.b
) result_t
order by result_t.a;
```

## 二十四、N 指标--累计去重

假设表 A 为事件流水表，客户当天有一条记录则视为当天活跃。

表 A：

```
   time_id          user_id
2018-01-01 10:00:00    001
2018-01-01 11:03:00    002
2018-01-01 13:18:00    001
2018-01-02 08:34:00    004
2018-01-02 10:08:00    002
2018-01-02 10:40:00    003
2018-01-02 14:21:00    002
2018-01-02 15:39:00    004
2018-01-03 08:34:00    005
2018-01-03 10:08:00    003
2018-01-03 10:40:00    001
2018-01-03 14:21:00    005
```

假设客户活跃非常，一天产生的事件记录平均达千条。

# 问题一：累计去重

## 输出结果如下所示：

日期　　当日活跃人数　　月累计活跃人数_截至当日

date_id　user_cnt_act　user_cnt_act_month

2018-01-01　　2　　2

2018-01-02　　3　　4

2018-01-03　　3　　5

## 参考答案：

```
SELECT  tt1.date_id
       ,tt2.user_cnt_act
       ,tt1.user_cnt_act_month
FROM
(   -- ④ 按照 t.date_id 分组求出 user_cnt_act_month，得到 tt1
 SELECT  t.date_id
        ,COUNT(user_id) AS user_cnt_act_month
 FROM
 (   -- ③ 表 a 和表 b 进行笛卡尔积，按照 a.date_id,b.user_id 分组，保证截止到当日的用户唯一，得出表 t。
  SELECT  a.date_id
         ,b.user_id
  FROM
  (   -- ① 按照日期分组，取出 date_id 字段当主表的维度字段 得出表 a
   SELECT  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd') AS date_id
   FROM test.temp_tanhaidi_20211213_1
   GROUP BY  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd')
  ) a
  INNER JOIN
  (   -- ② 按照 date_id、user_id 分组，保证每天每个用户只有一条记录，得出表 b
   SELECT  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd') AS date_id
          ,user_id
   FROM test.temp_tanhaidi_20211213_1
   GROUP BY  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd')
            ,user_id
  ) b
  ON 1 = 1
  WHERE a.date_id >= b.date_id
  GROUP BY  a.date_id
           ,b.user_id
 ) t
 GROUP BY  t.date_id
) tt1
```

```
LEFT JOIN
(    -- ⑥ 按照 date_id 分组求出 user_cnt_act，得到 tt2
 SELECT  date_id
        ,COUNT(user_id) AS user_cnt_act
 FROM
 (    -- ⑤ 按照日期分组，取出 date_id 字段当主表的维度字段  得出表 a
  SELECT  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd') AS date_id
         ,user_id
  FROM test.temp_tanhaidi_20211213_1
  GROUP BY  from_unixtime(unix_timestamp(time_id),'yyyy-MM-dd')
           ,user_id
 ) a
 GROUP BY date_id
) tt2
ON tt2.date_id = tt1.date_id
```

## 最后

第一时间获取最新大数据技术，尽在公众号：五分钟学大数据

搜索公众号：五分钟学大数据，学更多大数据技术！

其他大数据技术文档可下方扫码关注获取：