

```

1 create table user_level_repurchase(
2     `user_level` string,
3     `sku_id` string,
4     `AtLeastOnce` bigint,
5     `AtLeastTwice` bigint,
6     `Repurchase_rate` decimal(10,2),
7     `rank` int
8 )
9 row format delimited fields terminated by '\t'
10 location '/warehouse/gmall/ads/user_level_purchase/'

```

用户等级	商品id	复购率	排行
1	432534	0.9	1
1	21342	0.8	2
1	34	0.7	3
1	4543	0.6	4
1	6564	0.5	5
1	41234	0.4	6
1	7654	0.3	7
1	45637	0.2	8
1	23452134	0.1	9
1	23143	0.0	10

--1、最终一定是每个**用户等级** 显示**十条数据**，每条数据里包含 不同的 **商品信息、复购率、排行**，

--2、复购率排行：每个**用户等级内部** 按复购率排行，所以要开窗口，**窗口内部 按用户等级分区**，区内按复购率排序

--3、复购率：购买该商品次数至少2次的人数 / 购买该商品次数至少1次的人数

--4、所以要求出 **每个用户** 针对**每件商品**的 购买次数。---->>用户购买商品明细表。

4、一个月内 每个用户 针对每件商品的 购买次数。

```
1  (  
2  select  
3      sdd.user_level user_level,  
4      sdd.user_id user_id,  
5      sdd.sku_id sku_id,  
6      sum(sdd.order_count) order_count  
7      --在当前月内，用户对单一商品总购买次数  
8  from  
9      dws_sale_detail_daycount sdd  
10 where  
11     date_format('2019-02-10','yyyy-MM')=date_format(dt,'yyyy-MM')  
12 group by  
13     sdd.user_level,  
14     sdd.user_id,  
15     sdd.sku_id  
16 ) table1
```

3、复购率： 购买该商品次数至少两次的人数 / 购买该商品次数至少1次的人数

```
1  (  
2  select  
3      user_level,  
4      sku_id,  
5      sum(if(order_count>=1,1,0)) AtLeastOnce,  
6      sum(if(order_count>=2,1,0)) AtLeastTwice,  
7      --复购率： 重复购买多次的人数 与 至少购买了一次的人数 的比率。  
8      sum(if(order_count>=2,1,0))/sum(if(order_count>=1,1,0)) Repurchase_rate  
9  from  
10     table1  
11 group by  
12     user_level,  
13     sku_id  
14 )table2
```

2、排行：每个用户等级内部按复购率对商品排行，所以要开窗口，窗口内部按用户等级分区，区内按复购率排序

```

1  (
2  select
3      sku_id,
4      user_level,
5      AtLeastOnce,
6      AtLeastTwice,
7      Repurchase_rate,
8      rank() over(partition user_level order by Repurchase_rate desc) rank
9  from
10     table2
11 )table3

```

-- 1、挑选排行前十的

```

1  select
2      sku_id,
3      user_level,
4      AtLeastOnce,
5      AtLeastTwice,
6      Repurchase_rate,
7      rank
8  from
9      table3
10 where
11     rank <=10;

```

组成总表:

```

1  select
2      t3.user_level,
3      t3.sku_id,
4      t3.AtLeastOnce,
5      t3.AtLeastTwice,
6      t3.Repurchase_rate,
7      t3.rank
8  from
9      (
10     select
11         t2.sku_id sku_id,
12         t2.user_level user_level,
13         t2.AtLeastOnce AtLeastOnce,
14         t2.AtLeastTwice AtLeastTwice,
15         t2.Repurchase_rate Repurchase_rate,
16         rank() over(partition by t2.user_level order by t2.Repurchase_rate desc) rank
17     --也可以用row_number() 函数
18     from
19         (

```

```

19     select
20         t1.user_level user_level,
21         t1.sku_id sku_id,
22         sum(if(order_count>=1,1,0)) AtLeastOnce,
23         sum(if(order_count>=2,1,0)) AtLeastTwice,
24         --3、复购率： 重复购买多次的人数 与 至少购买了一次的人数 的比率。
25         sum(if(order_count>=2,1,0))/sum(if(order_count>=1,1,0)) Repurchase_rate
26     from
27     (
28         select
29             sdd.user_level user_level,
30             sdd.user_id user_id,
31             sdd.sku_id sku_id,
32             sum(sdd.order_count) order_count --用户对单一商品总购买次数 在当前月内。
33         from
34             dws_sale_detail_daycount sdd
35         where
36             date_format('2019-02-10','yyyy-MM')=date_format(dt,'yyyy-MM')
37         group by
38             sdd.user_level,
39             sdd.user_id,
40             sdd.sku_id
41     )t1
42     group by
43         t1.user_level,
44         t1.sku_id
45     )t2
46
47 ) t3
48 where
49     rank <=10 ;

```

```

1  #!/bin/bash
2
3  # 定义变量方便修改
4  APP=gmall
5  hive=/opt/module/hive/bin/hive
6
7  # 如果是输入的日期按照取输入日期；如果没输入日期取当前时间的前一天
8  if [ -n "$1" ] ;then
9      do_date=$1
10  else
11      do_date=`date -d "-1 day" +%F`
12  fi
13
14  sql="
15  insert into table "$APP".user_level_repurchase
16  select
17      t3.user_level,
18      t3.sku_id,
19      t3.AtLeastOnce,

```

```

20     t3.AtLeastTwice,
21     t3.Repurchase_rate,
22     t3.rank,
23     '$do_date'
24 from
25 (
26     select
27         t2.sku_id sku_id,
28         t2.user_level user_level,
29         t2.AtLeastOnce AtLeastOnce,
30         t2.AtLeastTwice AtLeastTwice,
31         t2.Repurchase_rate Repurchase_rate,
32         rank() over(partition by t2.user_level order by t2.Repurchase_rate desc) rank
--也可以用row_number() 函数
33     from
34     (
35         select
36             t1.user_level user_level,
37             t1.sku_id sku_id,
38             sum(if(order_count>=1,1,0)) AtLeastOnce,
39             sum(if(order_count>=2,1,0)) AtLeastTwice,
40             --3、复购率： 重复购买多次的人数 与 至少购买了一次的人数 的比率。
41             sum(if(order_count>=2,1,0))/sum(if(order_count>=1,1,0)) Repurchase_rate
42         from
43         (
44             select
45                 sdd.user_level user_level,
46                 sdd.user_id user_id,
47                 sdd.sku_id sku_id,
48                 sum(sdd.order_count) order_count --用户对单一商品总购买次数 在当前月内。
49             from
50                 dws_sale_detail_daycount sdd
51             where
52                 date_format('$do_date', 'yyyy-MM')=date_format(dt, 'yyyy-MM')
53             group by
54                 sdd.user_level,
55                 sdd.user_id,
56                 sdd.sku_id
57         )t1
58         group by
59             t1.user_level,
60             t1.sku_id
61         )t2
62
63 ) t3
64 where
65     rank <=10 ;
66 "
67 $hive -e "$sql"

```

【用户购买商品明细表】是从何而来的？为什么要建【用户商品明细表】？

--需求三：品牌复购率

复购率的计算：

---> 某个品牌购买至少一次的人数，购买至少两次的人数 的比值

---> **每个用户**对某个品牌的购买次数

---> 每一个用户每次购买的品牌有什么（既包括用户详细信息，又包括购买的商品详情）

---> 建一个【用户购买明细表】

--建表字段：

-----用户信息： user_id,user_level,user_age,user_gender

-----商品详情： sku_id, sku_name, sku_tm_id, order_price,.....

-----购买详情： 该用户对该商品的购买次数order_count , 购买金额 order_amount

--三类信息分别从三个表中查询：

-----user_info , sku_info, order_detail

----- 通过order_detail 关联 用户表、 和商品表

```
1 create external table dws_sale_detail_daycount
2 (
3     user_id    string comment '用户 id',
4     sku_id     string comment '商品 Id',
5     user_gender string comment '用户性别',
6     user_age   string comment '用户年龄',
7     user_level string comment '用户等级',
8
9     order_price decimal(10,2) comment '商品价格',
10    sku_name    string comment '商品名称',
11    sku_tm_id   string comment '品牌id',
12    sku_category3_id string comment '商品三级品类id',
13    sku_category2_id string comment '商品二级品类id',
14    sku_category1_id string comment '商品一级品类id',
15    sku_category3_name string comment '商品三级品类名称',
16    sku_category2_name string comment '商品二级品类名称',
17    sku_category1_name string comment '商品一级品类名称',
18    spu_id     string comment '商品 spu',
19    sku_num    int comment '购买个数',
20
21    order_count string comment '该用户对该商品 当日下单单数',
22    order_amount string comment '该用户对该商品 当日下单金额'
23 ) COMMENT '用户购买商品明细表'
24 PARTITIONED BY (`dt` string)
25 stored as parquet
26 location '/warehouse/gmall/dws/dws_user_sale_detail_daycount/'
```

```
27 | tblproperties ("parquet.compression"="snappy");
```