**一、查找每个用户连续登陆的最大天数**

use girls

create table user\_date(

uid int(20),

login\_date date

);

insert into user\_date

VALUES(201,'20170101')

,(201,'20170102')

,(202,'20170102')

,(202,'20170103')

,(203,'20170103')

,(201,'20170104')

,(202,'20170104')

,(201,'20170105')

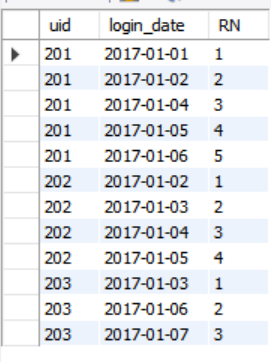
,(202,'20170105')

,(201,'20170106')

,(203,'20170106')

,(203,'20170107');

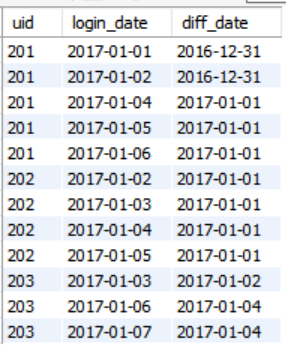
（1）第一步：按照uid分组，login\_date排序，增加一个新字段排名RN

select uid,login\_date,

row\_number() over (partition by uid order by login\_date) RN

from user\_date

（2）第二步：在第一步的基础上增加date\_diff列，即login\_date减去RN



select uid,login\_date,date\_sub(login\_date,interval RN day) as diff\_date

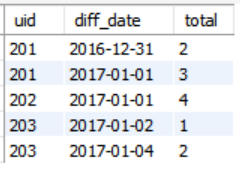
from(

select uid,login\_date,

row\_number() over (partition by uid order by login\_date) RN

from user\_date) t1

（3）得到每个人在每一天的连续登陆天数



select uid,diff\_date,count(1) as total

from(

select uid,login\_date,date\_sub(login\_date,interval RN day) as diff\_date

from(

select uid,login\_date,

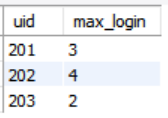
row\_number() over (partition by uid order by login\_date) RN

from user\_date) t1

) t2

group by uid,diff\_date

（4）取出最大值



select uid,max(total) as max\_login

from(

select uid,diff\_date,count(1) as total

from(

select uid,login\_date,date\_sub(login\_date,interval RN day) as diff\_date

from(

select uid,login\_date,

row\_number() over (partition by uid order by login\_date) RN

from user\_date) t1

) t2

group by uid,diff\_date

) t3

group by uid

（5）补充：login\_date是start\_date

select uid,login\_date,date\_sub(login\_date,interval RN day) diff\_date,count(1) as total

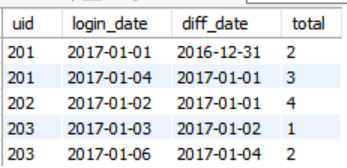
from(

select uid,login\_date,

row\_number() over (partition by uid order by login\_date) RN

from user\_date) t1

group by uid,diff\_date;



**二、选出每科排名前2名的同学名字**

use girls;

create table student\_test\_info(

id int(20),

student\_id int(20),

student\_name varchar(20),

course\_id int(20),

course\_name varchar(20),

score int(20)

);

insert into student\_test\_info

values(1,201001,"张黛丝",101,"商务统计",73),

(2,201001,"张黛丝",104,"时间序列分析",67),

(3,201001,"张黛丝",105,"随机过程",52),

(4,201001,"李美妮",104,"时间序列分析",61),

(5,201001,"李美妮",102,"应用回归分析",63),

(6,201001,"孙纳德",101,"商务统计",87),

(7,201001,"孙纳德",107,"计量经济学",82);

(1).编写hive sql语句提取“不偏科的学生”名单。

其中“不偏科的学生”的定义是：该学生选的所有课程中，每一门课程的考试分数均高于该课程的平均分。查询的环境为原生hive2.0

Select student\_name,course\_name,score,avg\_score

From(

Select course\_id,avg(score) avg\_score

From student\_test\_info

Group by course\_id

) a

inner join student\_test\_info b on a.course\_id=b.course\_id

Where score>avg\_score;

（2）选出每科排名前2名的同学名字

Select student\_name,course\_name,score,RN

from(

Select \*,dense\_rank() over (partition by course\_id order by score desc) RN

From student\_test\_info) as t1

Where RN=1 or RN=2;

**三、计算累计访问次数**

Select Item,name,score

From(

Select \*,dense\_rank() over (partition by Item order by score desc) as RN

from score\_detail

) as t1

Where RN=1 or RN=2;



Select name,

Sum(case when Item=”语文” then score else 0 end) as ‘Chinese’

Sum(case when Item=”数学” then score else 0 end) as ‘Math’

From score\_detail

Group by name;

**添加数据**

SELECT \* FROM girls.user\_date;

use girls;

create table action(

userid int(20),

visitdate date,

visitcount int(20)

);

insert into action

VALUES(001,'20170121',5)

,(002,'20170123',6)

,(003,'20170122',8)

,(004,'20170120',3)

,(001,'20170123',6)

,(001,'20170221',8)

,(002,'20170123',6)

,(001,'20170222',4);

**第一步：**

select userid "用户id",substr(visitdate,1,7) '月份',sum(visitcount) "小计"

from action

group by userid,substr(visitdate,1,7)

order by userid,substr(visitdate,1,7);



**第二步：使用窗口函数累计求和**

select userid,`month`,`count`,

sum(`count`) over (partition by userid order by `month`) as "累计"

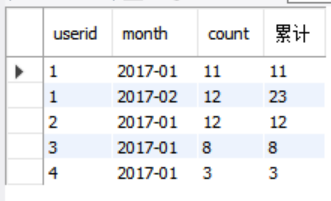
from(

select userid, substr(visitdate,1,7) as `month`, sum(visitcount) as `count`

from action

group by userid,substr(visitdate,1,7)

order by userid,substr(visitdate,1,7)) t1;



**四、计算用户留存率**

create table user\_date(

uid int(20),

login\_date date

);

insert into user\_date

VALUES(201,'20170101')

,(201,'20170102')

,(202,'20170102')

,(202,'20170103')

,(203,'20170103')

,(201,'20170104')

,(202,'20170104')

,(201,'20170105')

,(202,'20170105')

,(201,'20170106')

,(203,'20170106')

,(203,'20170107');

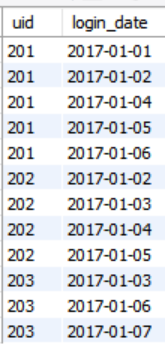
第一步：提取uid和login\_date并排序

select uid,login\_date

from user\_date

group by uid,login\_date

order by uid,login\_date;



第二步：增加一列first\_day,储存每个用户id最早登录时间

t2表是每个用户的最早登录日期

select t1.uid,login\_date,first\_day

from(

select uid,login\_date

from user\_date

group by uid,login\_date

order by uid,login\_date) t1

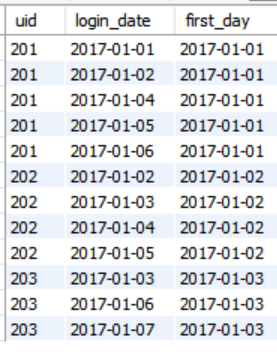
inner join (

select uid,min(login\_date) first\_day

from user\_date

group by uid) t2

on t1.uid=t2.uid



补充：由于表太长，先弄一个视图

create view login

as

select t1.uid,login\_date,first\_day

from(

select uid,login\_date

from user\_date

group by uid,login\_date

order by uid,login\_date) t1

inner join (

select uid,min(login\_date) first\_day

from user\_date

group by uid) t2

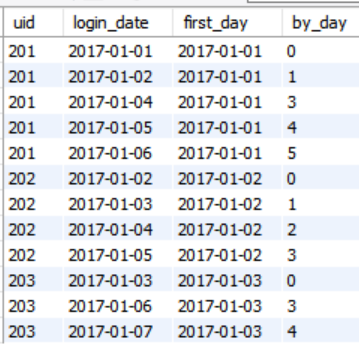
on t1.uid=t2.uid;

select \* from login;

**第三步：用登录时间-最早登录时间得到一列by\_day，这就得到了某一天登录离第一次登录有多长时间。**

select uid,login\_date,first\_day,datediff(login\_date,first\_day) as by\_day

from login



**第四步：提取字段作为列名**

select first\_day,

sum(case when by\_day=0 then 1 else 0 end) day\_0,

sum(case when by\_day=1 then 1 else 0 end) day\_1,

sum(case when by\_day=2 then 1 else 0 end) day\_2,

sum(case when by\_day=3 then 1 else 0 end) day\_3,

sum(case when by\_day=4 then 1 else 0 end) day\_4,

sum(case when by\_day=5 then 1 else 0 end) day\_5

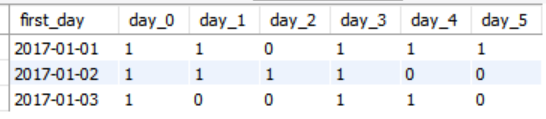
from(

select uid,login\_date,first\_day,datediff(login\_date,first\_day) as by\_day

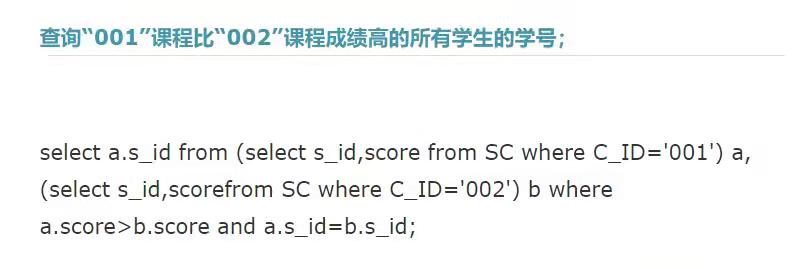
from login) a

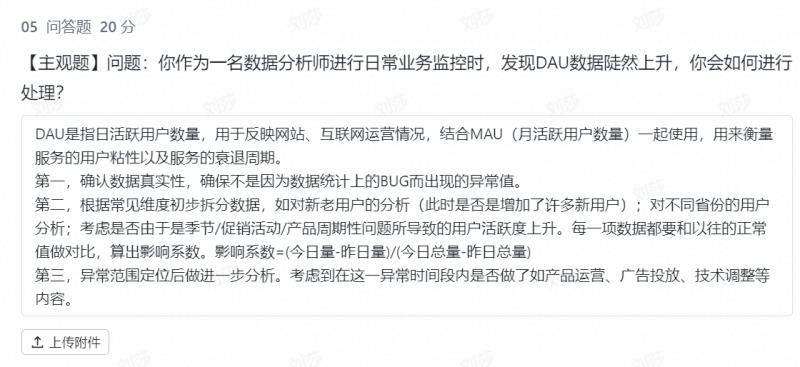
group by first\_day

order by first\_day



Day\_0是当日新增用户，从day\_1开始是每日活跃用户数量





**五、滴滴笔试**

****

**第一题：**

select uid

from

(select uid,amount from order where product\_id=001) a

(select uid,amount from order where product\_id=002) b

where a.amount>b.amount and a.uid=b.uid

第一题

Select uid,001product,002product

From

(Select uid,sum(

Case when

Product\_id=’001’ then amount

Else 0

end) as 001product,

Sum(

Case when

Product\_id=’002’ then amount

Else 0

end) as 002product

) a

Where 001product>002product

**第二题：**

Select a.name,avg(amount) as `平均申购金额`

From

(Select uid,name from users) a

Inner join

(select uid,amount) b

On a.uid=b.uid

Group by a.name

Having avg(amount)>100000 分组后筛选

**第三题：**

Select a.uid,a.name,count(product\_id) `购买产品数`,sum(amount) `购买总金额`

From

(Select uid,name from users) a

Inner join

(select uid,product\_id,amount) b

On a.uid=b.uid

**第四题：**

Select name,product\_id,amount,amount\_rank

From

(Select

name,product\_id,amount,row\_number() over(partition by product\_id order by amount desc) as amount\_rank

from

(Select name,product\_id,amount

From

(Select uid,name from users) a

Inner join

(select uid,product\_id,amount) b

On a.uid=b.uid) t1) t2

Where amount\_rank in (1,2,3)

**第五题：**

create table order\_1(

uid int(20),

order\_time date);

insert into order\_1

VALUES(201,'20170201')

,(201,'20170402')

,(202,'20170302')

,(202,'20170103')

,(203,'20170303')

,(201,'20170404')

,(202,'20170104')

,(201,'20170305')

,(202,'20170105')

,(201,'20170206')

,(202,'20170104')

,(201,'20170705')

,(202,'20170605')

,(201,'20170506')

,(203,'20170406')

,(203,'20170206')

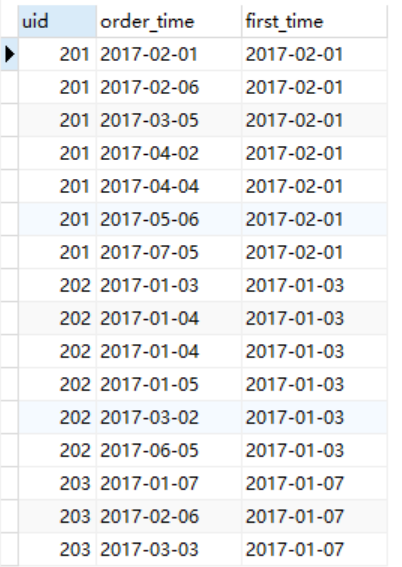
,(203,'20170107');

select \*

from order\_1

order by uid,order\_time;

**第一步：**得到每个用户的最早购买时间



select b.uid,b.order\_time,first\_time

from

(select uid,order\_time,min(order\_time) as first\_time

from order

group by uid) a

inner join

(select uid,order\_time

from order) b

on a.uid=b.uid

order by uid,order\_time;

**第二步：得到相差的月份**

select uid,order\_time,first\_time,DATEDIFF(mm,order\_time,first\_time) as diffmonth

from

(

select b.uid,b.order\_time,first\_time

from

(select uid,order\_time,min(order\_time) as first\_time

from order

group by uid) a

inner join

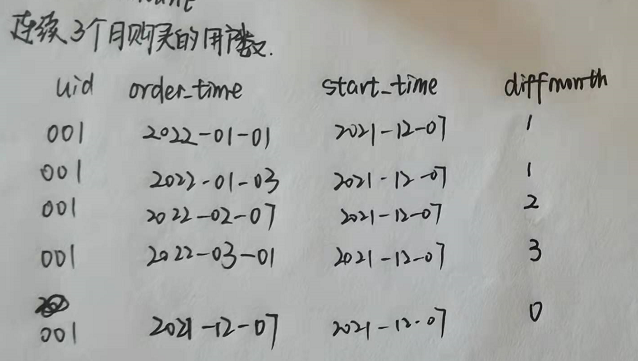
(select uid,order\_time

from order) b

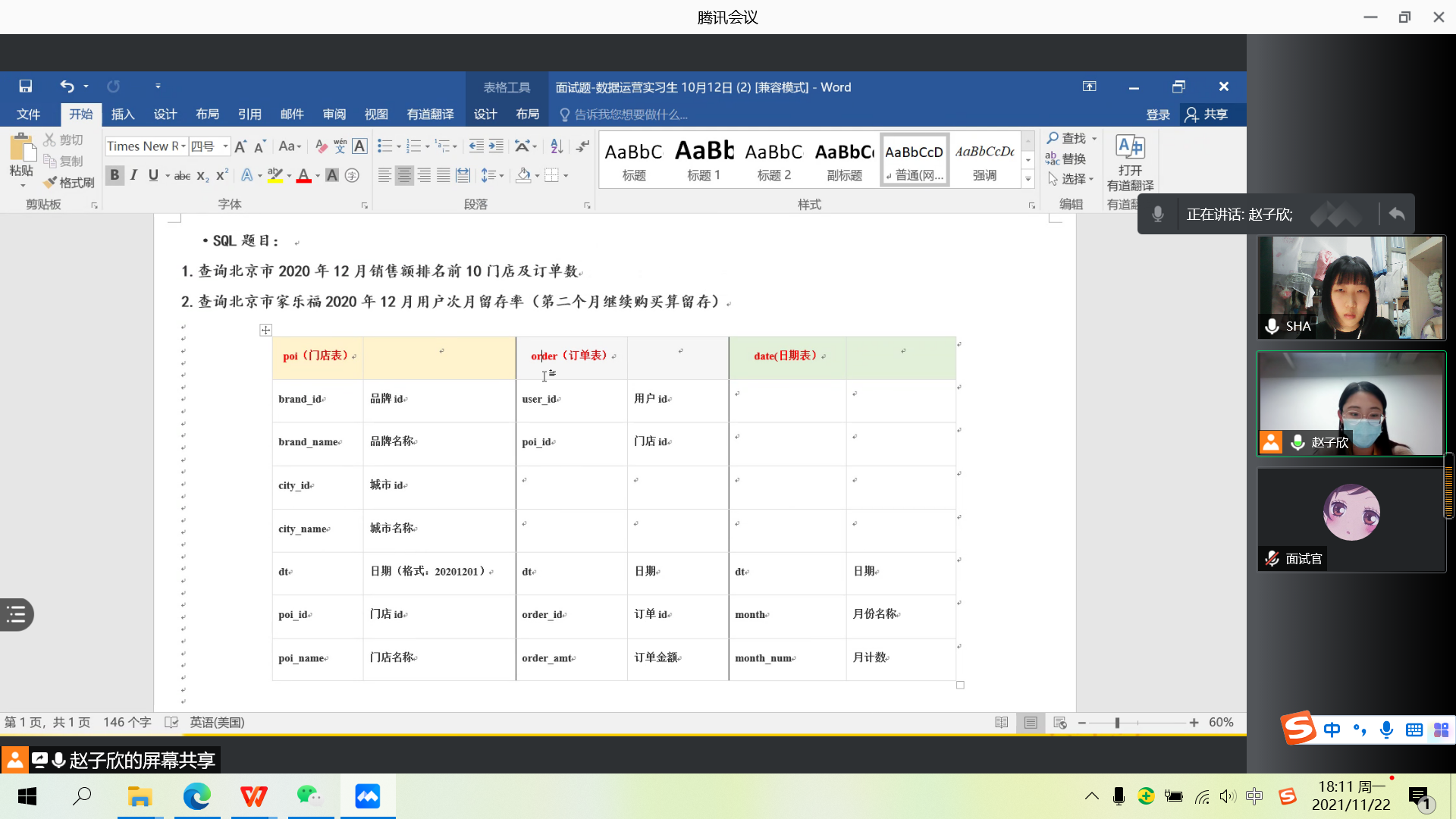
on a.uid=b.uid

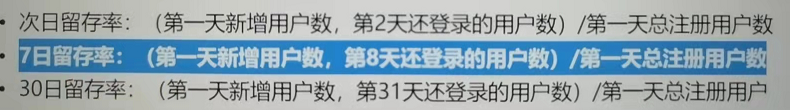
order by uid,order\_time

) t1



用户的diffmonth中1,2,3都出现，得到该条件下的用户数





今日注册的就是今日新增的，先得到每个用户的最早登录时间