

干锋好程序员大数据学院-练习题

制作人	版本	备注
李亚东	v1.0	hive场景题更新到25道题，所有题都重要。
李亚东	v4.0	hive场景题更新到35道题。注:所有题均为重点,难度均为中等难度,要求所有人必须会。
李亚东	V45.0	hive场景题更新到43道题。

1 hive场景题

1、了解哪些窗口函数，都是什么功能？找一个在某个业务中的应用？

手写窗口函数及功能意义，同时随便写一个带窗口函数的sql，并说明其sql的含义。

2、求出每个栏目的被观看次数及累计观看时长？

数据：

vedio表

Uid channl min

```
1 1 23
2 1 12
3 1 12
4 1 32
5 1 342
6 2 13
7 2 34
8 2 13
9 2 134
```

```
create table video(
  Uid int,
  channel string,
  min int
)
row format delimited
fields terminated by ' '
;
load data local inpath '/hivedata/video.txt' into table video;
```

```
select
channel,
count(1) num,
sum(min) total
from video
group by channel
;
```

3、编写sql实现

数据:

```
userid,month,visits
A,2015-01,5
A,2015-01,15
B,2015-01,5
A,2015-01,8
B,2015-01,25
A,2015-01,5
A,2015-02,4
A,2015-02,6
B,2015-02,10
B,2015-02,5
A,2015-03,16
A,2015-03,22
B,2015-03,23
B,2015-03,10
B,2015-03,1
```

每个用户截止到每月为止的最大单月访问次数和累计到该月的总访问次数，结果数据格式如下:

<u>用户</u>	<u>月份</u>	<u>最大访问次数</u>	<u>总访问次数</u>	<u>当月访问次数</u>
A	2015-01	33	33	33
A	2015-02	33	43	10
A	2015-03	38	81	38
B	2015-01	30	30	30
B	2015-02	30	45	15
B	2015-03	44	89	44

```
select
userid,
month,
visits,
max(visits) over(distribute by userid sort by month) maxvisit,
sum(visits) over(distribute by userid sort by month) totalvisit
from
(select
userid,
month,
sum(visits) visits
from visits
group by userid,month) t1
;
```

```
create table visits(
```

```

userid int,
month string,
visits int
)
row format delimited
fields terminated by ','
;

load data local inpath '/hivedata/visits.txt' overwrite into table visits;

```

4、编写连续7天登录的总人数：

数据：

t1表

Uid dt login_status(1登录成功,0异常)

1	2019-07-11	1
1	2019-07-12	1
1	2019-07-13	1
1	2019-07-14	1
1	2019-07-15	1
1	2019-07-16	1
1	2019-07-17	1
1	2019-07-18	1
2	2019-07-11	1
2	2019-07-12	1
2	2019-07-13	0
2	2019-07-14	1
2	2019-07-15	1
2	2019-07-16	0
2	2019-07-17	1
2	2019-07-18	0
3	2019-07-11	1
3	2019-07-12	1
3	2019-07-13	1
3	2019-07-14	1
3	2019-07-15	1
3	2019-07-16	1
3	2019-07-17	1
3	2019-07-18	1

```

create table login(
  uid int,
  dt string,
  login_status int
)
row format delimited
fields terminated by ' '
;

load data local inpath '/hivedata/login.txt' into table login;

select
  uid,

```

```

dt
from
(select
t1.uid uid,
date_sub(t1.dt,t1.rm) dt
from
(select
uid,
dt,
row_number() over(distribute by uid sort by dt) rm
from login
where
login_status=1) t1)t2
group by uid
having count(uid) >7
;

```

5、你知道的排名函数有哪些？说一说它们之间的区别？

文字说明即可

6、编写sql语句实现每班前三名，分数一样并列，同时求出前三名按名次排序的一次的分差：

数据：
stu表

Stu_no	class	score
1	1901	90
2	1901	90
3	1901	83
4	1901	60
5	1902	66
6	1902	23
7	1902	99
8	1902	67
9	1902	87

编写sql实现，结果如下：

结果数据：

班级	stu_no	score	rn	rn1	rn_diff
1901	1	90	1	1	90
1901	2	90	2	1	0
1901	3	83	3	1	-7
1902	7	99	1	1	99
1902	9	87	2	2	-12
1902	8	67	3	3	-20

```

create table stu(
Stu_no int,
class string,
score int
)

```

```

row format delimited
fields terminated by '\t'
;

load data local inpath '/hivedata/stu.txt' into table stu;

select
class,
stu_no,
score,
dr,
score-nvl(lag(score) over(distribute by class sort by dr),0)
from
(select
class,
stu_no,
score,
dense_rank() over(distribute by class sort by score desc) dr
from stu) t1
where t1.dr<4
;

```

7、对于行列互换，你有哪些解决方式，详细说明每一种方式？

使用语言描述即可

8、编写sql实现行列互换：

数据如下：

id	userid	subject	score
1	001	语文	90
2	001	数学	92
3	001	英语	80
4	002	语文	88
5	002	数学	90
6	002	英语	75.5
7	003	语文	70
8	003	数学	85
9	003	英语	90
10	003	政治	82

编写sql实现，得到结果如下：

userid	语文	数学	英语	政治	total
001	90	92	80	0	262
002	88	90	75.5	0	253.5
003	70	85	90	82	327
total	248	267	245.5	82	842.5

```
create table score(
id int,
sid string,
subject string,
score int
)
row format delimited
fields terminated by ','
;

load data local inpath '/hivedata/score.txt' into table score;

select
sid,
sum(case when subject="语文" then score else 0 end) as `语文`,
sum(case when subject="数学" then score else 0 end) as `数学`,
sum(case when subject="英语" then score else 0 end) as `英语`,
sum(case when subject="政治" then score else 0 end) as `政治`
from score
group by sid
;
```

9、编写sql实现如下：

数据：
t1表
uid tags
1 1,2,3
2 2,3
3 1,2

编写sql实现如下结果：

uid tag
1 1
1 2
1 3
2 2
2 3
3 1
3 2

```
create table market(
id int,
storage string,
allocation string,
outdt string
)
```

```

row format delimited
fields terminated by '\t'
;

load data local inpath '/hivedata/market.txt' into table market;


uid tags
1  1,2,3
2  2,3
3  1,2


create table t1(
uid int,
tags string
)
row format delimited
fields terminated by '\t'
;
load data local inpath '/hivedata/t1.txt' into table t1;

select
uid,
tag
from t1
lateral view explode(split(tags",")) t2 as tag
;

```

10、行转列

数据:

T1表:

Tags
1,2,3
1,2
2,3

T2表:

Id	lab
1	A
2	B
3	C

根据T1和T2表的数据，编写sql实现如下结果:

ids	tags
1,2,3	A,B,C
1,2	A,B
2,3	B,C

```

create table t2(
tags string
)
;
load data local inpath '/hivedata/t2.txt' overwrite into table t2;

```

```

create table t3(
  id int,
  lab string
)
row format delimited
fields terminated by ' '
;
load data local inpath '/hivedata/t3.txt' overwrite into table t3;

select
  tags,
  concat_ws(",",collect_list(lab))
from
  (select
    t1.tags tags,
    t3.lab lab
  from
    (select
      tags,
      id
    from t2
    lateral view explode(split(tags",")) tmp as id) t1
  join t3
  on t1.id=t3.id) tmp
group by tags
;

```

11、行转列

数据:

t1表:

id	tag	flag
a	b	2
a	b	1
a	b	3
c	d	6
c	d	8
c	d	8

编写sql实现如下结果:

id	tag	flag
a	b	1 2 3
c	d	6 8

```

create table t4(
  id string,
  tag string,
  flag int
)
row format delimited
fields terminated by '\t'
;
load data local inpath '/hivedata/t4.txt' overwrite into table t4;

```



```

select
id,
tag,
concat_ws("|",collect_set(flag)) flag
from t4
group by id,tag
;

```

12、列转行

数据:

t1表

uid	name	tags
1	goudan	chihuo,huaci
2	mazi	sleep
3	laotie	paly

编写sql实现如下结果:

uid	name	tag
1	goudan	chihuo
1	goudan	huaci
2	mazi	sleep
3	laotie	paly

```

create table t5(
uid string,
name string,
tags string
)
row format delimited
fields terminated by '\t'
;
load data local inpath '/hivedata/t5.txt' overwrite into table t5;

select
uid,
name,
tag
from t5
lateral view explode(split(tags,",")) t1 as tag
;

```

13、行转列

数据:

t1表:

uid	contents
1	i love china
2	china is good i i like

统计结果如下,如果出现次数一样,则按照content名称排序:

content	cnt
i	3

```
china    2
good     1
like     1
love     1
is       1
```

```
create table content(
uid int,
contents string
)
row format delimited
fields terminated by '\t'
;
load data local inpath '/hivedata/content.txt' overwrite into table content;

select
ct,
count(1) cn
from
(select
ct
from content
lateral view explode(split(contents,"\\|")) tmp as ct) tmp
group by ct
order by cn desc,ct
;
```

14、列转行

数据:

t1表

id	course
1	a
1	b
1	c
1	e
2	a
2	c
2	d
2	f
3	a
3	b
3	c
3	e

根据编写sql, 得到结果如下(表中的1表示选修, 表中的0表示未选修):

id	a	b	c	d	e	f
1	1	1	1	0	1	0
2	1	0	1	1	0	1
3	1	1	1	0	1	0

```
create table course(
id int,
course string
)
```

```

row format delimited
fields terminated by ','
;
load data local inpath '/hivedata/course.txt' overwrite into table course;

select
id,
sum(case when c.course="a" then 1 else 0 end) as `a`,
sum(case when c.course="b" then 1 else 0 end) as `b`,
sum(case when c.course="c" then 1 else 0 end) as `c`,
sum(case when c.course="d" then 1 else 0 end) as `d`,
sum(case when c.course="e" then 1 else 0 end) as `e`,
sum(case when c.course="f" then 1 else 0 end) as `f`
from course c
group by id
;

```

15、时间戳函数：unix_timestamp, from_unixtime

获取当前时间戳：

获取"2019-07-31 11:57:25"对应的时间戳：

获取"2019-07-31 11:57"对应的时间戳：

获取时间戳:1564545445所对应的日期和时分秒：

获取时间戳:1564545446所对应的日期和小时(yyyy/mm/dd HH)：

```
select unix_timestamp("2019-07-31 11:57:25","yyyy-mm-dd HH:MM:ss")
```

```
select unix_timestamp("2019-07-31 11:57","yyyy-mm-dd HH:MM")
```

```
select from_unixtime(1564545445,"yyyy-mm-dd HH:MM:ss");
```

```
select from_unixtime(1564545445,"yyyy/mm/dd HH");
```

16、时间格式转换：yyyyMMdd -> yyyy-MM-dd

数据：

t1表

20190730

20190731

编写sql实现如下的结果：

2019-07-30

2019-07-31

```

create table dt(
dt string
);

load data local inpath '/hivedata/dt.txt' overwrite into table dt;

select
from_unixtime(unix_timestamp(dt,"yyyymmdd"),"yyyy-mm-dd" )
from dt
;

```

17、

数据:

店铺,月份,金额

```

a,01,150
a,01,200
b,01,1000
b,01,800
c,01,250
c,01,220
b,01,6000
a,02,2000
a,02,3000
b,02,1000
b,02,1500
c,02,350
c,02,280
a,03,350
a,03,250

```

编写Hive的HQL语句求出每个店铺的当月销售额和累计到当月的总销售额?

```

create table login_time(
log_time string,
uid string
)
row format delimited
fields terminated by ','
;
load data local inpath '/hivedata/login_time.txt' overwrite into table
login_time;

select
uid,
max(cn) cn
from
(select
uid,
count(uid) cn
from
(select
uid uid,
date_sub(l_time,dr) dt

```

```

from
(select
distinct
from_unixtime(unix_timestamp(log_time,"yyyy-mm-dd HH:MM:ss"),"yyyy-mm-dd" )
l_time,
uid,
dense_rank() over(distribute by uid sort by
from_unixtime(unix_timestamp(log_time,"yyyy-mm-dd HH:MM:ss"),"yyyy-mm-dd" )) dr
from login_time) t1) t2
group by uid,dt) t3
group by uid
;

```

18、Hive是否发生过数据倾斜，怎么处理的，原理是什么？

使用语言表述即可。

19、Hive中什么时候使用过array和map，为什么使用？

语言描述即可

20、使用sql编写如下图的需求：

2.1 最大连续访问天数

Table A 是一个用户登陆时间记录表，当月每次登陆一次会记录一条记录，A表如下：

log_time	uid
2018-10-01 12:34:11	123
2018-10-02 13:21:08	123
2018-10-02 14:08:09	456
2018-10-04 05:10:22	123
2018-10-04 21:38:38	456
2018-10-05 09:57:32	123
2018-10-06 13:22:56	123

需计算出每个用户本月最大连续登录天数。如表 A 样例数据中，用户 123 最大连续登录天数为 3 而用户 456 最大连续登录天数为 1

-- 注意：可能需要对原始数据做清洗，保证每个用户每天只有一条登录信息

```

select uid, max(cnt)
  from (
select uid, substr(login_time, 1, 7), count(*) cnt
  from (
select uid, login_time
      ,date_sub(login_time, row_number() over (partition by uid order by
login_time)) as groupflag
  from (

```

```

select uid, substring(login_time, 1, 10) login_time
from db_exer.exercise20
group by uid, substring(login_time, 1, 10)
)
)
group by uid, groupflag, substr(login_time, 1, 7)
)
group by uid

```

21、使用sql实现如下：

样例数据：

t1表

gender,cookie,ip,timestamp,ua

F,1707041428491566106,111.200.195.186,1208524973899,Dalvik%2F2.1.0%20%28Linux%3B%20U%3B%20Android

...具体数据如下图

1、根据下面的日志数据（1）进行数据统计

```

fields:cookie,ip,timestamp,ua
,1707041428491566106,111.200.195.186,1208524973899,Dalvik%2F2.1.0%20%28Linux%3B%20U%3B%20Android%2
,1708301506121297324,111.198.237.214,1506123358316,MobileTrackingDemo%2F1%20CFNetwork%2F811.5.4%20E
,1708151725391824724,111.200.195.186,1509113859105,MobileTrackingDemo%2F1%20CFNetwork%2F811.5.4%20E
,1708151725391824724,111.200.195.186,1511356333356,MobileTrackingDemo%2F1%20CFNetwork%2F811.5.4%20E
,1707041428491566106,111.200.195.186,1208524973899,MobileTrackingDemo%2F1%20CFNetwork%2F811.5.4%20E

```

- 使用awk 统计 不同天的pv , uv
- 如果数据量有10亿条，使用什么方法进行统计？
 - 计算维度数据：分小时，分天，分地域的 pv, uv
 - 如何通过ip解析地域
 - urlencode 如何处理
- 如果有样本数据(如下)，如何通过样本数据（2）和 日志数据（1）算出某个地域的F数据统计，例如，北京地域下 F性别 20岁的人数有多少？

样本数据（2）格式如下：

```

cookie,gender,age
1707041428491566106,F,20
1708151725391824724,M,30

```

- 可以使用的方方法有？
- 是否有什么优化的可能？

将图片中的awk修改为使用sql编写，然后将上述题作出回答？

统计pv/uv的使用sql，其它问题语言描述即可。

22、使用hive求出两个数据集的差集？

数据

t1表：

id name

1 zs

2 ls

t2表:

id name

1 zs

3 ww

结果如下:

id name

2 ls

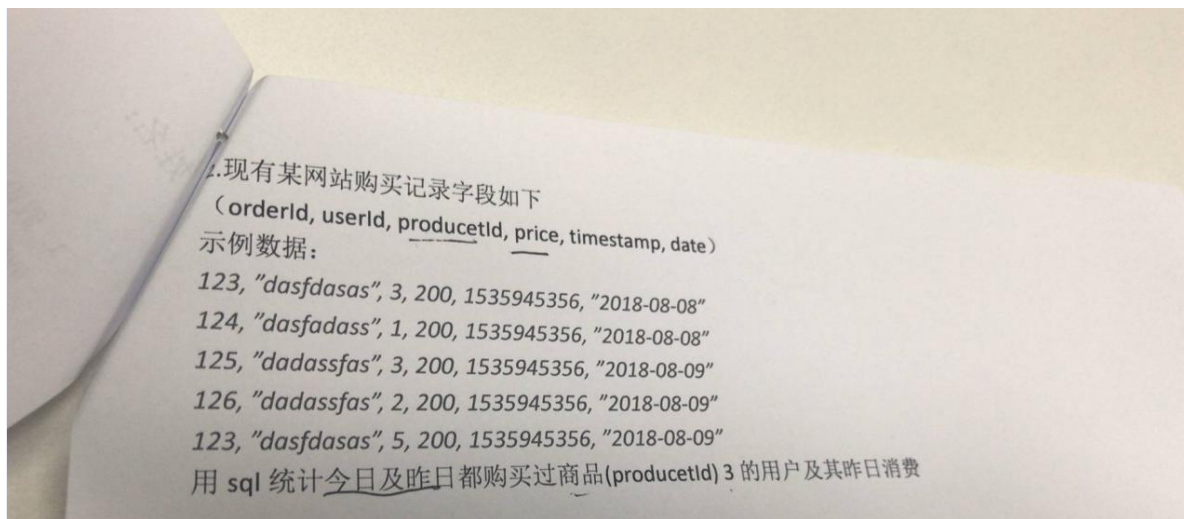
3 ww

```
create table diff_t1(
id string,
name string
)
row format delimited
fields terminated by ' '
;
load data local inpath '/hivedata/diff_t1.txt' overwrite into table diff_t1;

create table diff_t2(
id string,
name string
)
row format delimited
fields terminated by ' '
;
load data local inpath '/hivedata/diff_t2.txt' overwrite into table diff_t2;

select t1.id as id,
t1.name as name
from diff_t1 t1
left join diff_t2 t2
on t1.id=t2.id
where t2.id is null
union
select
t2.id as id,
t2.name as name
from diff_t1 t1
right join diff_t2 t2
on t1.id=t2.id
where t1.id is null
;
```

23、使用hive的hql实现如下需求：



使用hive的hql实现如上需求？

24、使用hive的hql如下:

表 user_action_log 用户行为数据

U_id	Time	Action
1	Time1	Read
3	Time2	Comment
1	Time3	Share
2	Time4	Like
1	Time5	Write
2	Time6	Like
3	Time7	Write
2	Time8	Read

1. 分析用户行为习惯，找到每一个用户在表中的第一次行为

使用代码实现

25、每个用户连续登陆的最大天数？

数据:

login表

uid,date

1,2019-08-01

1,2019-08-02

1,2019-08-03

2,2019-08-01

2,2019-08-02

3,2019-08-01


```
3,2019-08-03
4,2019-07-28
4,2019-07-29
4,2019-08-01
4,2019-08-02
4,2019-08-03
```

结果如下:

```
uid cnt_days
1 3
2 2
3 1
4 3
```

26、请使用sql计算pv、uv？

数据:

t1表

```
uid date url
1 2019-08-06 http://www.baidu.com
2 2019-08-06 http://www.baidu.com
3 2019-08-06 http://www.baidu.com
3 2019-08-06 http://www.soho.com
3 2019-08-06 http://www.meituan.com
3 2019-08-06
```

结果如下:

```
date uv pv
2019-08-6 3 5
```

27、hive中coalesce()、nvl()、concat_ws()、collect_list()、collect_set()、regexp_replace().这几个函数的意义？

使用语言描述即可。

28、

3.有表如下记录了智智同学玩某 moba 游戏每局的英雄 pick 情况

Id	Names
1	亚索,挖掘机,艾瑞莉娅,洛,卡莎
2	亚索,盖伦,奥巴马,牛头,皇子
3	亚索,盖伦,艾瑞莉娅,宝石,琴女
4	亚索,盖伦,赵信,老鼠,锤石

请用 HiveSQL 计算出出场次数最多的 top3 英雄及其 pick 率 (=出现场数/总场数) :

使用hive的hql实现如上需求。

29、有三个表，分别是：

区域(district) 区域中有两个字段分别是区域ID(disid)和区域名称(disname)
城市(city) 城市有两个字段分别是城市ID(cityid)和区域ID(disid)
订单(order) 订单有四个字段分别是订单ID(orderid)、用户ID(userid)、城市ID(cityid)和消费金额(amount)。

district表:

disid	disname
-------	---------

1	华中
---	----

2	西南
---	----

city表:

cityid	disid
--------	-------

1	1
---	---

2	1
---	---

3	2
---	---

4	2
---	---

5	2
---	---

order表:

oid	userid	cityid	amount
-----	--------	--------	--------

1	1	1	1223.9
---	---	---	--------

2	1	1	9999.9
---	---	---	--------

3	2	2	2322
---	---	---	------

4	2	2	8909
---	---	---	------

5	2	3	6789
---	---	---	------

6	2	3	798
---	---	---	-----

7	3	4	56786
---	---	---	-------

8	4	5	78890
---	---	---	-------

高消费者是消费金额大于1w的用户，使用hive hql生成如下报表：

区域名	高消费者人数	消费总额
-----	--------	------

hql语句。

30、某APP每天访问数据存放在表access_log里面，包含日期字段ds,用户类型字段user_type，用户账号user_id,用户访问时间log_time,请使用hive的hql语句实现如下需求：

(1)、每天整体的访问UV、PV？

(2)、每天每个类型的访问UV、PV？

(3)、每天每个类型中最早访问时间和最晚访问时间？

(4)、每天每个类型中访问次数最高的10个用户？

31、一张大表A(上亿条记录)和小表B(几千条记录)，如果join出现数据倾斜，有什么解决办法？

使用语言描述即可。

32、有如下三张表:

表A(登录表):

ds	user_id
2019-08-06	1
2019-08-06	2
2019-08-06	3
2019-08-06	4

表B(阅读表):

ds	user_id	read_num
2019-08-06	1	2
2019-08-06	2	3
2019-08-06	3	6

表C(付费表):

ds	user_id	price
2019-08-06	1	55.6
2019-08-06	2	55.8

基于上述三张表, 请使用hive的hql语句实现如下需求:

- (1)、用户登录并且当天有个阅读的用户数, 已经阅读书籍数量
- (2)、用户登录并且阅读, 但是没有付费的用户数
- (3)、用户登录并且付费, 付费用户书籍和金额

33、hive的hql中, left outer join和left semi join的区别?

使用语言描述即可。

34、有一个订单表和渠道表, 结构如下:

有订单表和渠道表, 表结构如下:

```
create table order
order_id long,
user_id long, comment '用户 id',
amount double comment '订单金额',
channel string comment '渠道',
time string comment '订单时间  yyyy-MM-dd hh:mi:ss'
)
partitioned by (dt string comment '天, yyyy-mm-dd');
```

请使用hive hql查询出2019-08-06号 每个渠道的下单用户数、订单总金额。

hql语句实现, 结果表头如下:

channel	user_num	order_amount
---------	----------	--------------

35、考虑表设计和sql查询：

设计数据库表，用来存放学生基本信息，课程信息，学生的课程及成绩，并给出 sql 语句，查询平均成绩大于85的所有学生。

```
create table stu_1(  
  id string,  
  name string,  
  age int,  
  addr string  
)  
row format delimited  
fields terminated by ','  
;  
  
create table course_1(  
  cid string,  
  cname string  
)  
row format delimited  
fields terminated by ','  
;  
  
create table course_sc(  
  id string,  
  cid string,  
  score int  
)  
row format delimited  
fields terminated by ','  
;  
  
load data local inpath '/hivedata/course_1.txt' overwrite into table course_1;  
load data local inpath '/hivedata/stu_1.txt' overwrite into table stu_1;  
load data local inpath '/hivedata/course_sc.txt' overwrite into table course_sc;  
  
select  
  cs.id,  
  avg(score) avgscore  
from course_sc cs  
group by cs.id  
having avgscore>85  
;
```

36、需求如下：

一、有用户表 User(uid, name) 以及黑名单表 BanUser(uid)

1、用 left join 方式写 sql 查出所有不在黑名单的用户信息：

2、用 not exists 方式写 sql 查询所有不在黑名单的用户信息：

```
create table u(  
  id string,  
  name string  
)  
row format delimited  
fields terminated by ','  
;  
  
create table banuser(  
  id string  
)  
;  
load data local inpath '/hivedata/banuser.txt' overwrite into table banuser;  
load data local inpath '/hivedata/u.txt' overwrite into table u;  
  
select  
  u.id,  
  u.name  
from u  
left join banuser b  
on u.id=b.id  
where b.id is null  
;  
  
select  
  u.id,  
  u.name  
from u  
where not exists (select 1 from banuser b where u.id=b.id)  
;
```

37、需求如下：

二、有如下考试成绩表 course_score

id	userid	course	score	term
1	zhangsan	数学	80	2015
2	lisi	语文	90	2016
3	wangwu	化学	70	2017
4	zhangsan	语文	80	2015
5	zhangsan	化学	90	2015

编写 sql 完成如下查询，一次查询实现最好，也可写多次查询实现：

1、查出每个学期每门课程最高分记录（包含全部 5 个字段）。

2、查出单个学期中语文课在 90 分以上的学生的数学成绩记录（包含全部 5 个字段）。

course_score

1,zhangsan,数学,80,2015

2,lisi,语文,90,2016

3,lisi,数学,70,2016

4,wangwu,化学,80,2017

5,zhangsan,语文,85,2015

6,zhangsan,化学,80,2015

```
create table course_score(  
id string,  
name string,  
course string,  
score int,  
year string  
)  
row format delimited  
fields terminated by ','  
;
```

```
load data local inpath '/hivedata/course_score.txt' overwrite into table  
course_score;
```

```
select  
s.id,  
s.name,  
s.course,  
s.score,  
s.year  
from  
course_score s  
join
```

```
(select
course,
year,
max(score) score
from course_score
group by course,year) t1
on s.course=t1.course
and
s.year=t1.year
and
s.score=t1.score
;
```

```
select
s.id,
s.name,
s.course,
s.score,
s.year
from
course_score s
join
(select
id,
name,
course,
score,
year
from
course_score
where
score >=70
and
course="语文") t1
on s.name=t1.name
where s.course="数学"
;
```

38、需求如下

t1表:

name	course	score
aa	English	75
bb	math	85
aa	math	90

使用hql输出以下结果

name	English	math
aa	75	90
bb	0	85

39、需求如下

t1表:

用户 商品

A P1

B P1

A P2

B P3

请你使用hql变成如下结果:

用户 P1 P2 P3

A 1 1 0

B 1 0 1

40、需求如下

3

dpt	部门
dpt_id	dpt_name
1	产品
2	技术

User	用户表
User_id	Dpt_id
1	1
2	1
3	2
4	2
5	3

result		
User_id	Dpt_id	Dpt_name
1	1	产品
2	1	产品
3	2	技术
4	2	技术
5	3	其他部门

原数据表时user和dpt表, 请使用hql实现result表中的结果.

41、需求如下

t1表:

order_id	order_type	order_time
111	N	10:00
111	A	10:05
111	B	10:10

是用hql获取结果如下:

order_id	order_type_1	order_type_2	order_time_1	order_time_2
111	N	A	10:00	10:05
111	A	B	10:05	10:10

42、需求如下

t1表

name	sex	hobby
janson	男	打乒乓球、游泳、看电影
tom	男	打乒乓球、看电影

hobby最多3个值,使用hql实现结果如下:

name	sex	hobby1	hobby2	hobby3
janson	男	打乒乓球	游泳	看电影
tom	男	打乒乓球	看电影	

43、需求如下

表t1(注:数据时正常的访问日志数据,分隔符全是空格)

```
8.35.201.160 - - [16/May/2018:17:38:21 +0800] "GET /uc_server/data/avatar/000/01/54/22_avatar_middle.jpg HTTP/1.1" 200 5396
```

使用hive的hql实现结果如下:

ip	dt	url
8.35.201.160	2018-5-16	17:38:21 /uc_server/data/avatar/000/01/54/22_avatar_middle.jpg