1903班周考

1、请使用hive hql实现如下需求：

```
数据:
参见qianfeng.log文件。
数据格式如下:
phone        dt               site(基站)                              type(1:进站 0:出站)
18688888888,20160327082400,16030401EAFB68F1E3CDF819735E1C66,1
18611132889,20160327082500,16030401EAFB68F1E3CDF819735E1C66,1
18688888888,20160327170000,16030401EAFB68F1E3CDF819735E1C66,0
18611132889,20160327180000,16030401EAFB68F1E3CDF819735E1C66,0

需求如下(注:数据只进不出或者只出不进都将视为无效数据):
(1)、每个人(手机号)在每个基站停留的总时长?
(2)、每天每个基站的累计停留总时长的TOP3(每个人累计之和)?
```

```
1、
select phone,site,from_unixtime(sum(diff),'HHmmss') cnt
from
(
select *,unix_timestamp(dt,'yyyyMMddHHmmss') -
unix_timestamp(lagdt,'yyyyMMddHHmmss') diff
from
(
select *
,lag(dt,1) over(distribute by phone,site sort by dt) lagdt
,lag(type,1) lagtype
from t
) a
where a.type = 0 and a.type <> a.lagtype
) b
group by phone,site
;

select
select b.phone,b.site,a.dt adt,min(b.dt) bdt
from (
select *,row_number() over() from type = 1
) a
join
(
select *,row_number() over() from type = 0
) b on a.phone = b.phone and a.site = b.site
where b.dt > a.dt
group by b.phone,b.site,a.dt
;
```

2、请使用hive的hql实现如下需求：

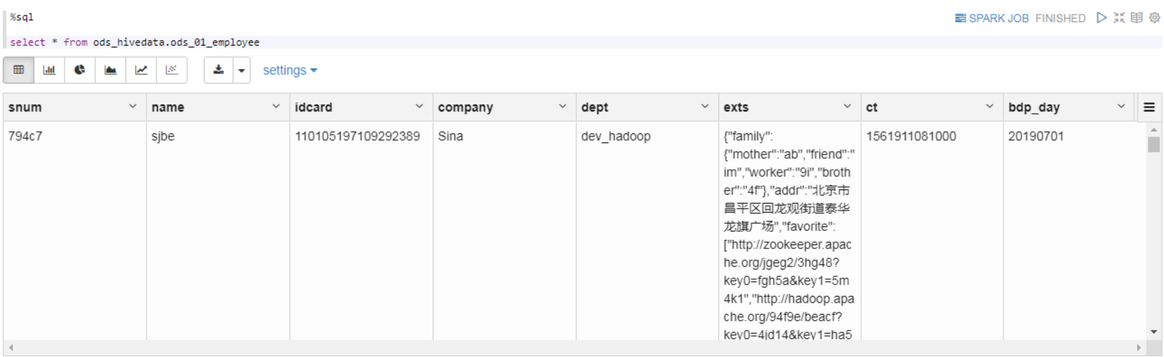
```
数据:
sp表
store   product sale
1001    A   600
1001    B   700
1002    A   300
1002    B   200
1003    A   800
1003    B   100

st表
store_id    store_name
1001    旺旺
1002    阿黄
1003    阿香

需求如下:
(1)、销售总和大于1000的店铺名称和销售总和？
select
store_id,
store_name,
sum(sp.sale) sumsale
from st
join sp on st.store_id = sp.store
group by
store_id,
store_name
having sum(sp.sale) > 1000
;

(2)、每个店铺的店铺名称和累计销售额？
select *,
sum(sale) over(distribute by store sort by rn )
from
(
select *,row_number() over(distribute by store) rn
) a
;

(3)、每个店铺、每个店铺和每个产品的销售额？
店铺、产品、销售额
基本聚合：
group by
店铺、产品....
高级聚合：
grouping sets：
指定组合方式（聚合方式）
cube：（2的N次方种组合）(维度之间没有关系)
立方体
rollup：（维度之间是有包含关系、依赖关系）
从右边依次减少一个维度

grouping_id：展示分组的层级

select store_name,product,sum(sale)
grouping_id
from t
group by
```

```
store_name,product
grouping sets((store_name,product),(store_name)
;

select store_name,product,sum(sale)
grouping_id
from t
group by
store_name,product
with cube /
with rollup
;

select store_name,product,sum(sale) as sumsale
from t
group by store_name,product
union all
select store_name,null as product,sum(sale) as sumsale
from t
group by store_name
union all
select null store_name,null as product,sum(sale) as sumsale
from t
union all
select null store_name,product,sum(sale) as sumsale
from t
group by product
;


(4)、每个店铺总销量、每个店铺销量降序排名和销量降序排名，结果如下:
store_name  product     sale    store_sale  store_num   sale_num
旺旺   A   600     1300    2   3
旺旺   B   700     1300    1   2
阿黄   A   300     500     1   4
阿黄   B   200     500     2   5
阿香   A   100     900     2   6
阿香   B   800     900     1   1

with tmp as
(
select *,
sum(sale) over(distribute by store_name) as store_sale,
row_number() over(distribute by store_name sort by sale desc) rn
from st
join sp on sp.store = st.store_id
),
tmp1 as (
select * from tmp
),
tmp2 as (
select * from tmp1
)
select *,row_number() over(sort by sale desc )
from tmp
;

多表输出
```

```
from (
select *,
sum(sale) over(distribute by store_name) as store_sale,
row_number() over(distribute by store_name sort by sale desc) rn
from st
join sp on sp.store = st.store_id
) tmp
insert  into t
select *
join a on
where
insert into
select
;
```

3、有贴源层数据，员工信息表如下，请统计大家最喜欢的top3技术（如有需要进行过滤，比如员工没有喜欢的技术）

```
create external table if not exists ods_hivedata.ods_01_employee(
    snum string comment '员工编号',
    name string comment '员工姓名',
    idcard string comment '员工身份证',
    company string comment '员工公司',
    dept string comment '员工部门',
    exts string comment '扩展信息(json格式)',
    ct bigint comment '创建时间'
  ) partitioned by (bdp_day string)
  stored as parquet
  location '/data/hivedata/ods/employee/'
```



```
%sql
select * from ods_hivedata.ods_01_employee
```

snum	name	idcard	company	dept	exts	ct	bdp_day
794c7	sjbe	110105197109292389	Sina	dev_hadoop	{"family":{"mother":"ab","friend":"im","worker":"9i","brother":"4f"},"addr":"北京市昌平区回龙观街道泰华龙旗广场","favorite":["http://zookeeper.apache.org/jgeg2/3hg48?key0=fgh5a&key1=5m4k1","http://hadoop.apache.org/94f9e/beacf?key0=4jd14&key1=ha5	1561911081000	20190701

其中exts为扩展信，形式如下

```
{
    "family": {
        "mother": "ab",
        "friend": "im",
        "worker": "9i",
        "brother": "4f"
    },
    "addr": "北京市昌平区回龙观街道泰华龙旗广场",
    "favorite": [
        "http://zookeeper.apache.org/jgeg2/3hg48?key0=fgh5a&key1=5m4k1",
        "http://hadoop.apache.org/94f9e/beacf?key0=4jd14&key1=ha53c",
        "http://hive.apache.org/f7bhb/81g8j?key0=kj53e&key1=8k634"
    ],
    "likes": [
        "hbase",
        "hbase",
        "elasticsearch",
        "mysql"
    ]
}
```

```
--重点在于json串的处理
--1、get_json_object
--2、json_tuple，需要和lateral view 结合使用
--3、获取到的likes内容包含[]，需要使用regexp_replace处理
--4、split
--5、lateral view explode处理数组，展开成行数据


select *
from
(
select regexp_replace(like,'"',''),
count(1) as cnt,
row_number() over(sort by count(1) desc) rn
from ods_hivedata.ods_01_employee
lateral view explode(split(regexp_replace(get_json_object(exts,"$.likes"), "[\\[\\]]", ''),',')) t as like
where bdp_day = '20190701'
group by like
) a
where a.rn < 4
;

select like,
count(1) as cnt
from ods_hivedata.ods_01_employee
lateral view explode(split(regexp_replace(get_json_object(exts,"$.likes"), "[\\[\\]]", ''),',')) t as like
where bdp_day='20190701'
group by like
order by cnt desc
limit 3
;

select *
from
(
select regexp_replace(like,'"',''),
count(1) as cnt,
row_number() over(sort by count(1) desc) rn
from (
select likes
from ods_hivedata.ods_01_employee
lateral view json_tuple(exts,"likes") t as likes
where bdp_day = '20190701'
)
lateral view explode(split(regexp_replace(likes, "[\\[\\]]", ''),',')) t as like
group by like
) a
where a.rn < 4
;
```

4、设计数据库表，用来存储用户基本信息，订单信息，订单商品信息，支付信息，商品信息，类别信息，并给出表结构信息，同时计算每天、不同产品、不同类别、不同平台的销售金额？

```
使用文字描述和sql来作答。

参考高效运营平台项目的业务表设计
```