

第三天笔记

第三天笔记

SQL练习

Hive数据类型

整型: TINYINT、SMALLINT、INT、BIGINT

浮点: FLOAT、DOUBLE

布尔类型: BOOL (False/True)

字符串: STRING

时间类型:

复杂数据类型:

Hive HQL

DDL

DQL

SQL练习

1、count(*)、count(1)、count('字段名') 区别

2、HQL 执行优先级:

from、where、group by、having、order by、join、select、limit

3、where 条件里不支持不等式子查询，实际上是支持 in、not in、exists、not exists

```
-- 列出与“SCOTT”从事相同工作的所有员工。
select  t1.EMPNO
        ,t1.ENAME
        ,t1.JOB
from emp t1
where t1.ENAME != "SCOTT" and t1.job in(
    select  job
    from emp
    where ENAME = "SCOTT");

7900,JAMES,CLERK,7698,1981-12-03,950,null,30
7902,FORD,ANALYST,7566,1981-12-03,3000,null,20

select  t1.EMPNO
        ,t1.ENAME
        ,t1.JOB
from emp t1
where t1.ENAME != "SCOTT" and exists(
    select  job
    from emp t2
    where ENAME = "SCOTT"
    and t1.job = t2.job
);
```

4、hive中大小写不敏感

5、在hive中，数据中如果有null字符串，加载到表中的时候会变成 null（不是字符串）

如果需要判断 null, 使用 某个字段名 is null 这样的方式来判断

或者使用 nvl() 函数, 不能 直接 某个字段名 == null

6、使用explain查看SQL执行计划

```
explain select  t1.EMPNO
              ,t1.ENAME
              ,t1.JOB
from emp t1
where t1.ENAME != "SCOTT" and t1.job in(
    select  job
    from emp
    where ENAME = "SCOTT");

# 查看更加详细的执行计划, 加上extended
explain extended select  t1.EMPNO
                      ,t1.ENAME
                      ,t1.JOB
from emp t1
where t1.ENAME != "SCOTT" and t1.job in(
    select  job
    from emp
    where ENAME = "SCOTT");
```

Hive数据类型

整型: TINYINT、SMALLINT、INT、BIGINT

浮点: FLOAT、DOUBLE

布尔类型: BOOL (False/True)

字符串: STRING

时间类型:

- 时间戳 timestamp
- 日期 date

```
create table testDate(
    ts timestamp
    ,dt date
) row format delimited fields terminated by ',';

// 2022-02-19 14:24:57.200,2022-02-19
```

- 时间戳与时间字符串转换

```
// from_unixtime 传入一个时间戳以及pattern (yyyy-MM-dd) 可以将 时间戳转换成对应格式的字符串
select from_unixtime(1630915221,'yyyy年MM月dd日 HH时mm分ss秒')

// unix_timestamp 传入一个时间字符串以及pattern, 可以将字符串按照pattern转换成时间戳
select unix_timestamp('2021年09月06日 16时00分21秒','yyyy年MM月dd日 HH时mm分ss秒');
select unix_timestamp('2021-01-14 14:24:57.200')
```

复杂数据类型:

- array

```
create table testArray(  
    name string,  
    weight array<string>  
)row format delimited  
fields terminated by '\t'  
COLLECTION ITEMS terminated by ',';  
  
select name,weight[0] from testArray;
```

杨老板 140,160,180
张志凯 160,200,180

- map

key:value,key2:v2,k3:v3

```
create table scoreMap(  
    name string,  
    score map<string,int>  
)ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
COLLECTION ITEMS TERMINATED BY ','  
MAP KEYS TERMINATED BY ':';  
  
select name,score['语文'] from scoreMap;
```

小明 语文:91,数学:110,英语:40
小红 语文:100,数学:130,英语:140

- struct

```
create table scoreStruct(  
    name string,  
    score struct<course:string,score:int,course_id:int,teacher:String>  
)ROW FORMAT DELIMITED  
FIELDS TERMINATED BY '\t'  
COLLECTION ITEMS TERMINATED BY ',';  
  
select name,score.course,score.score from scoreStruct;
```

小明 语文,91,000001,余老师
小红 数学,100,000002,体育老师

<https://blog.csdn.net/woshixuye/article/details/53317009>

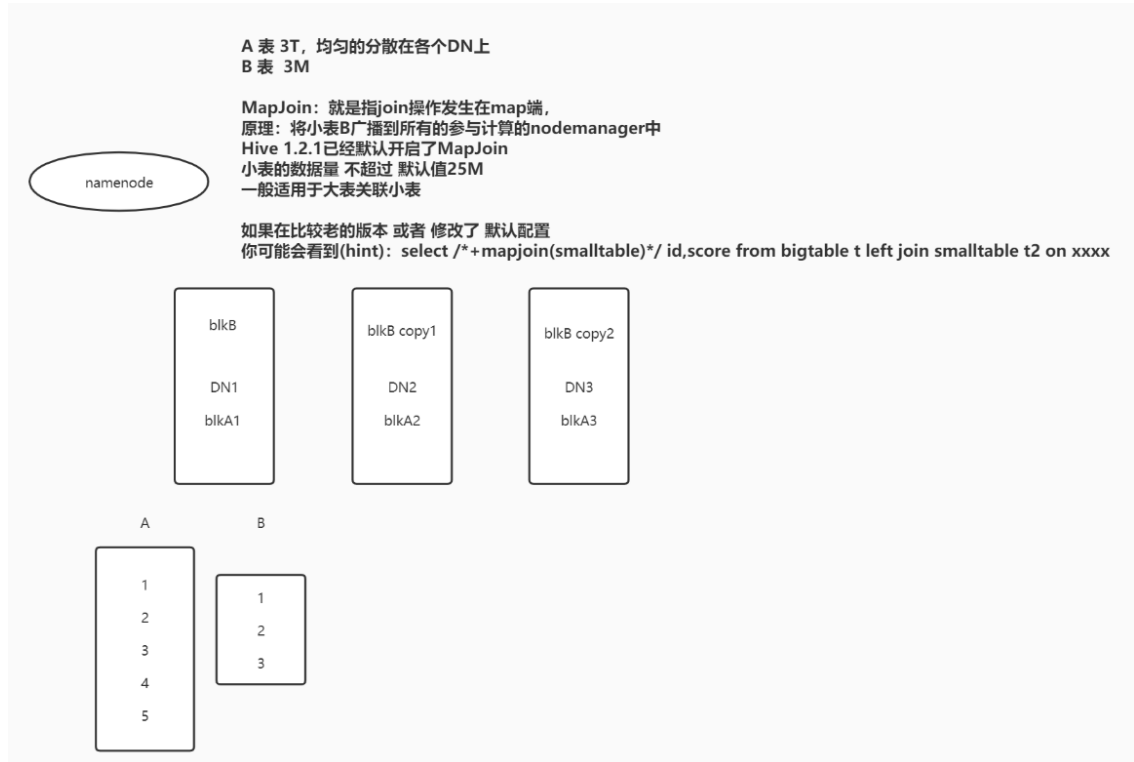
Hive HQL

DDL

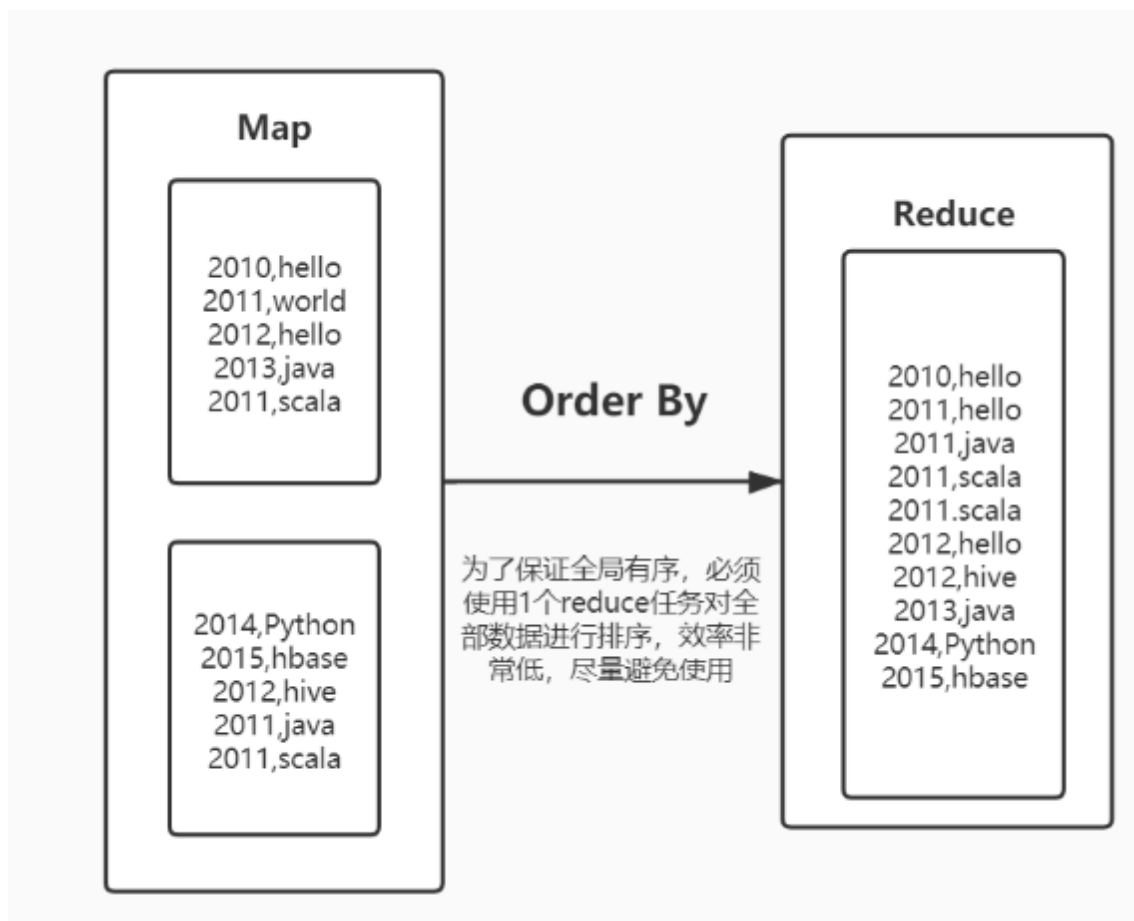
DQL

```
select id,name from tb t where ... and .... group by xxx having xxxx order by xxx  
asc/desc limit n;
```

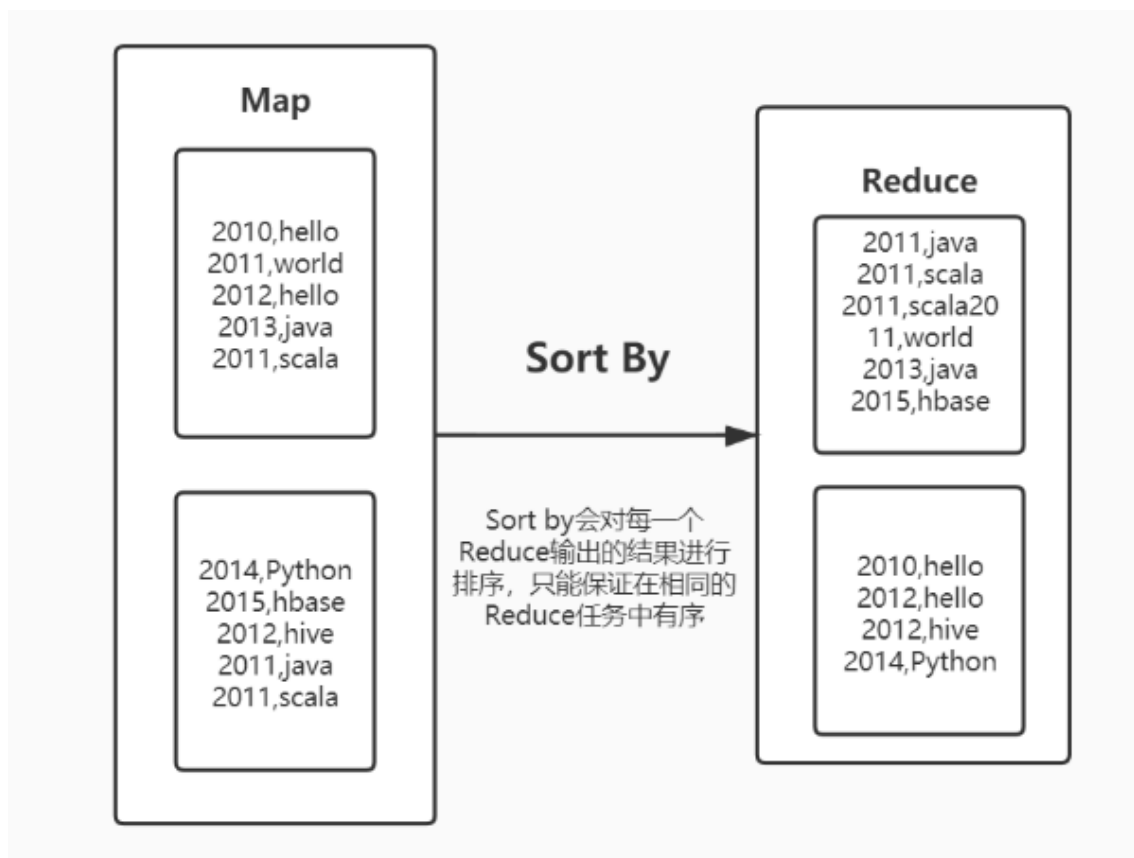
- where: 过滤数据、!!! 分区裁剪!!!
- join: left join、right join、join 注意MapJoin



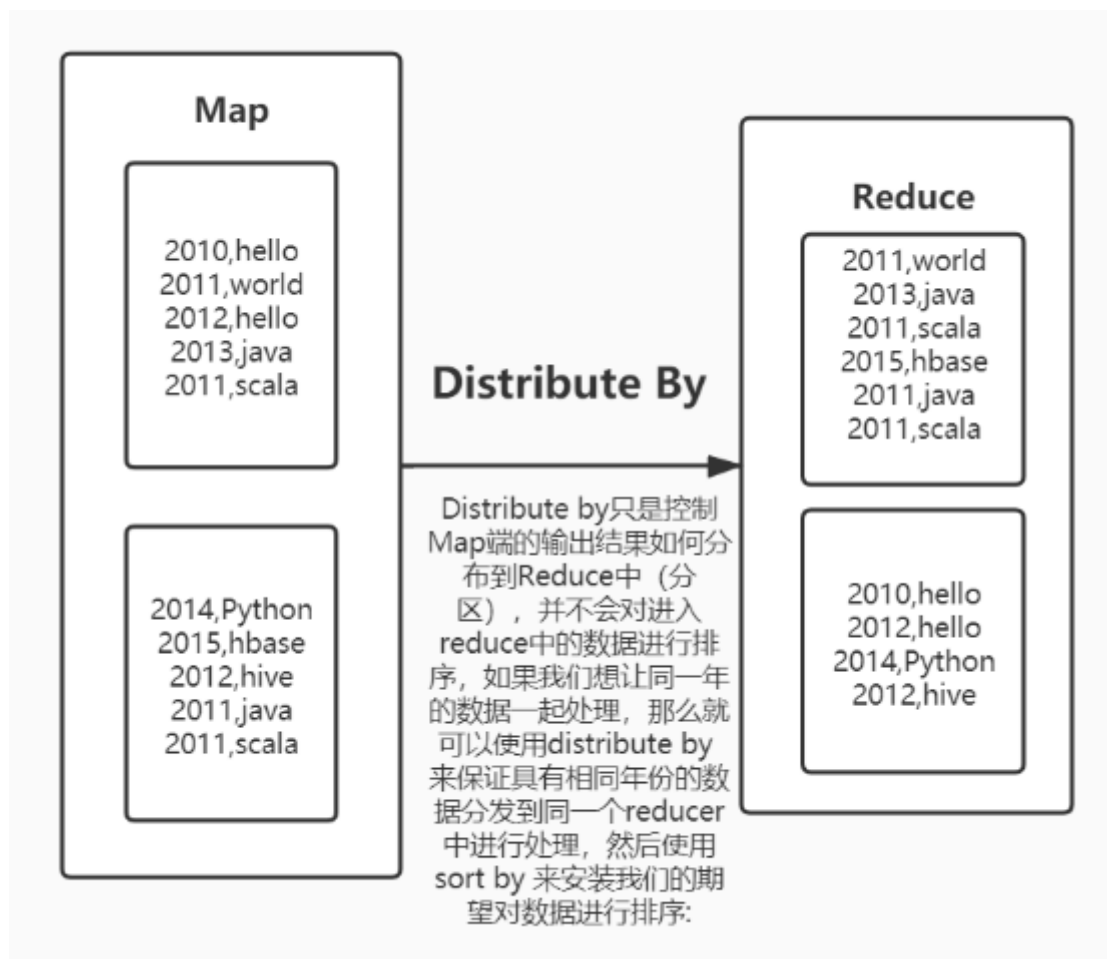
- group by: 通常结合聚合函数一起使用
- order by: 全局排序



- sort by: 局部排序



- distribute by: 分区



- cluster by

