# mdrill sql使用手册

目录

[mdrill sql使用手册 1](#_Toc362363362)

[mdrill的分区 2](#_Toc362363363)

[查询明细 2](#_Toc362363364)

[统计汇总 2](#_Toc362363365)

[分类汇总 2](#_Toc362363366)

[mdrill的过滤 3](#_Toc362363367)

[JDBC查询(依赖的jar包请去lib下找) 3](#_Toc362363368)

修改记录

2013年7月23日 子落 创建文档

## mdrill的分区

mdrill的设计默认是使用分区的，也是按照分区进行存储的，除非强制使用single类型的分区外，查询的时候必须指定分区。

目前mdrill的分区字段为thedate,格式为yyyyMMdd

在顶层SQL的where条件中必须有如下三种分区设定的一种

1. thedate=’yyyyMMdd’ 直接指定某一个分区
2. thedate in (yyyyMMdd, yyyyMMdd, yyyyMMdd) 给出一系列日期
3. thedate >= yyyyMMdd and thedate<= thedate 给出一个范围

## 查询明细

1. 通过mdrill可以查询top 1万条数据的明细，举例如下

select ipv,price from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 limit 0,20

1. 对于明细的数据，可以进行排序，也就是order by

select ipv,price from rpt\_hitfake\_auctionall\_d where thedate>=‘20130201’ and thedate<=20130202 order by price desc limit 0,20

## 统计汇总

mdrill目前支持sum,max,min,count,dist五种汇总函数,

1. dist就是sql中的count(distinct(xxx)),但是采用的是近似计算，会有一定的误差

具体实现原理，请参考<https://github.com/muyannian/higo/wiki/distinct>

1. count我们有另种使用方法

第一种是:count(列名)，针对具体某一列进行count统计，当然如果该列值如果存在NULL值，不会作为count计数。

第二种是: count(\*),即使存在NULL的列，也会列入计数。

我们给几个常见的统计的使用离子

select count(clickcount0),sum(clickcount0) from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 limit 0,1

select count(clickcount0),count(\*) from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 limit 0,1

## 分类汇总

1. mdrill目前支持多维分类汇总统计，也就是sql中的group by

举例如下：

select thedate,count(thedate) as cnt,sum(clickcount0) from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 group by thedate limit 0,20

1. 另外分类汇总后，mdrill可以按照某一列的值进行排序，如果分类汇总后的组数小于1万组，为准确排序，如果超过1万组则为近似的排序和计算，有可能存在排序的顺序和计算的结果不正确的情况

针对超过1W组后获取的汇总数值有可能不准确的情况，业务方要在第一次查询过后，根据返回的数据，进行第二次查询，用以获取准确的结果，当然之后的mdrill我们会逐步实现这些。

使用示例如下：

select thedate,count(thedate) as cnt,sum(clickcount0) as sum from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 group by thedate order by sum desc limit 0,20

## mdrill的过滤

1. 目前mdrill的过滤支持如下几种

=:等于

<>：不等于

>=:大于等于

<=:小于等于

>:大于

<:小于

in:属于列表

not in:属于列表

like:模糊匹配

1. 目前mdrill的过滤表达式 为如下的格式

分区过滤 and 条件一 and 条件二 and ….条件N

也就是说最外层 必须有分区过滤 并且只能是and的关系

但是每个条件里面可以是用括号嵌套的关系，

比如说

分区过滤 and 条件一 and (条件二 or 条件三 or (条件四 and 条件五))

给出一个我们经常使用的例子

select category\_level3\_name,count(higoempty\_count\_l) from rpt\_hitfake\_auctionall\_d where thedate>='20130201' and thedate<=20130202 and category\_level3\_name like '暗黑%' group by category\_level3\_name limit 0,100

## JDBC查询(依赖的jar包请去lib下找)

mdrill的SQL支持JDBC的方式，需要先启动监控与jdbc服务，启动服务的方法如下  
nohup ./bluewhale mdrillui 1107 ../lib/adhoc-web-0.18-beta.war >ui.log &

String connstr = "jdbc:mdrill://localhost:1107";  
Class.forName("com.alimama.mdrill.jdbc.MdrillDriver");  
Connection con = DriverManager.getConnection(connstr, "", "");  
Statement stmt = con.createStatement();  
HigoQueryResultSet res = null;  
res = (HigoQueryResultSet) stmt.executeQuery("select ipv,price from rpt\_hitfake\_auctionall\_d where thedate = '20130201' limit 0,20");  
System.out.println("totalRecords:"+res.getTotal());  
List<String> colsNames = res.getColumnNames();  
for (int i = 0; i < colsNames.size(); i++) {  
    System.out.print(colsNames.get(i));  
    System.out.print("\t");  
}  
System.out.println();  
while (res.next()) {  
    for (int i = 0; i < colsNames.size(); i++) {  
        System.out.print(res.getString(colsNames.get(i)));  
        System.out.print("\t");  
    }  
    System.out.println();  
}  
con.close();