阿里中间件挑战赛

# 赛题描述

天猫与淘宝上每天产生了大量的交易，每一笔交易会生成一条交易记录。现在有许多交易记录文件，文件中每一行视作一条交易记录，每条记录中可能包含若干Key-Value对，这样的key-value对称为属性。文件内的交易记录之间是无序的。你的任务是将这些小文件组织成有序的形式并提供查询接口，供业务方查询。  
更多赛题的详细信息请参见： <https://code.aliyun.com/MiddlewareRace/order-system>。

**交易信息：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 交易ID | 商品ID | 买家ID | 交易时间 | 红包金额 | 交易金额 |
| 10000 | 1001 | 101 | 1460552595 | 0.2 | 188 |
| 10001 | 1001 | 102 | 1460532598 | 0.5 | 188 |
| 10002 | 1001 | 101 | 1460532500 | 0.6 | 138 |
| 10003 | 1002 | 102 | 1460532607 |  | 300 |
| 10004 | 1003 | 103 | 1460532608 |  | 288 |
| 10005 | 1001 | 104 | 1460520604 | 0.7 | 20 |

**买家信息：**

|  |  |  |  |
| --- | --- | --- | --- |
| 买家ID | 姓名 | 收货地址 | 联系电话 |
| 101 | 张三 | 杭州市 | 888886 |
| 102 | 李四 | 北京市 | 888887 |
| 103 | 王五 | 上海市 | 233-12345 |
| 104 | 孙六 | 广州市 | 88888 |
| 105 | 杨七 | 香港 | 45678 |

**商品信息：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 商品ID | 名称 | 卖家ID | 标价 | 打折价格 |
| 1001 | 洗发水 | 301 | 188 | 138 |
| 1002 | 沐浴露 | 301 | 300 |  |
| 1003 | 牙膏 | 302 | 20 |  |

**业务方需要选手支持下面几种查询方法：**1、提供交易ID，查询某次交易的某些属性，比如查询某次交易的收货地址以及电话。查询的字段可能不在交易记录中，需要到商品信息或者买家信息中查询。  
2、查询某位买家某个时间范围内的所有交易信息，比如查询买家在2016.4.14~2016.4.15之间所有交易的交易ID与交易金额。输出结果按交易时间从大到小排列  
3、查询某位卖家的某个商品的全部交易订单，输出结果按交易订单从小到大排序  
4、对某个商品的所有交易信息进行求和，比如查询某个卖家商品的折扣券金额总和

# 程序目标

你的coding目标是实现一个继承自OrderSystem接口的OrderSystemImpl类。 实现一些简单的订单查询功能。

construct接口负责将数据文件路径传送到OrderSystemImpl中，其中storeFolders每个路径包含一块存储设备。用于供你存储内存外的数据。

queryOrder、queryOrdersByBuyer、queryOrdersBySaler、sumOrdersByGood提供按照特定方式查询数据的接口，具体规则可以查看OrderSystem.java的注释部分

数据格式

有三种类型的文件，分别为订单文件、商品文件、买家文件。文件的基本格式相同。 我们对文件格式的约定如下

## 数据格式

* 数据都存储在文件中，每行代表一条记录。
* 每种类型的文件可能包含复数个文件。
* 保证所有文件的记录中的主键不重复。
* 每条记录包含若干Key-Value对。Key又称作字段。
* 每条记录的字段保证不重复。
* Key-Value对之间用制表符'\t'分隔，内容中如果出现空格，把空格当作Key或者Value的内容
* Key为ASCII字符集中的a..zA..Z0..9\_-
* Value可以为Boolean, Long, Double, String之中的任意类型，原始数据中不会声明类型
* 如果类型为Boolean，其对应的字符是'true'和'false'
* 记录之间不保证顺序
* 记录内的字段之间也不保证数据
* key的长度不超过256字节，value长度不超过64K

可以参考目录中的样例数据：buyer\_records.txt、good\_records.txt、order\_records.txt

每种类型的文件中有一些字段是一定存在的，且类型固定。查询操作主要通过这些字段进行：

交易信息文件一定包含的字段:

orderid - 订单号，类型为Long

buyerid - 买家ID，类型为String

goodid － 商品ID，类型为String

createtime - 交易创建时间，类型为Long

商品信息文件一定包含的字段:

goodid - 商品ID，类型为String

salerid - 卖家ID，类型为String

买家信息文件一定包含的字段:

buyerid - 买家ID，类型为String

**提示**

数据量上，商品文件和买家文件总大小大于4G，订单文件总大小小于100G。

数据中的某些字段比较稀有，在所有记录中出现较少。某些字段比较常见，在记录中经常出现

## 查询条件

一共有四种类型的查询：

queryOrder，查询指定订单号的交易信息，可以指定要查询的列

queryOrdersByBuyer，查询买家某个时间范围内的交易信息

queryOrdersBySaler，查询卖家某个商品的交易信息，可以指定要查询的列

sumOrdersByGood，对商品的某个字段进行求和

**提示**

1. queryOrder和queryOrdersBySaler两种查询大部分情况下只查询少量字段
2. 查询可能存在热点
3. 测试程序会并发进行查询，选手程序要保证查询接口能支持并发调用

## 程序打包

程序的类名一定为OrderSystemImpl，并实现一个不带参数的OrderSystemImpl构造函数，提供public访问权限。 Pom配置请参考git中的demo项目OrderSystemImpl，使用预定义的打包插件。

## 第二/三方库规约

仅允许依赖JavaSE 7 包含的lib

## 测试环境

JDK 1.7

多核CPU (暂定6核)

jvm 4G内存

可以使用空间充足的多块磁盘(暂定3块)

# 评分标准

选手程序的运行分为两个阶段，准备阶段与查询阶段。

1、在准备阶段，选手的程序有至多一小时的时间进行数据排序和整理成选手认为最佳的形式，以满足查询的需要。如果超时，则认为程序失败，不计算分数。

2、在查询阶段，测试程序会按照一定的查询操作序列调用程序的查询接口。程序输出结果并比对标准结果，如果结果错误，则认为程序失败，不计算分数。如果一小时内未完成所有查询，那么查询成功的次数会被记录为选手的分数。

如果两阶段均在一小时内完成，那么两阶段的总耗时将会被记录为总分从少到多排名。如果第二阶段未在一小时内完成所有查询，那么完成查询的次数会被记录为分数从高到低并排名，但排名会低于完成第二阶段所有查询的选手。

# 设计方案

## 布隆过滤器

布隆过滤器的特点是会出现误报，但是不会漏报。

## 一致性哈希

queryOrder，

查询指定订单号的交易信息，可以指定要查询的列

queryOrdersByBuyer，

查询买家某个时间范围内的交易信息

queryOrdersBySaler，

查询卖家某个商品的交易信息，可以指定要查询的列

sumOrdersByGood，

对商品的某个字段进行求和

对于第一个查询queryOrder，目前的想法是先使用布隆过滤器测试订单是否存在，不存在直接返回null;若存在则到文件中查询，然后这一步，打算预处理时将订单使用哈希算法分文件存储，这样查询阶段使用哈希算法定位到订单所在的小文件，到里面去查找相应的订单

#　mdw

恩，需要需要考虑用那些字段，最理想的感觉是按卖家id或（买家id）来分,但是文件数太多是否会有影响？按日期分应该比较好控制，但是范围不太清楚，可能要采用除一个数余几来确定。我现在念头是这样的：按时间分第一层，第二层数据重复一份？买家id 分一份，卖家id分一份。