数据结构final project

一、实验要求

本次实验要求实现一个简单的数据库，支持单点以及区间的增删改查等操作，具体要求功能见第三部分。实验要求使用合适的数据结构，尽量提高操作效率，最终的评分将基于操作的正确性和效率两个方面。本实验中禁止使用c++ STL容器(<array>,<deque>,<forward_list>,<list>,<map>,<queue>,<set>,<stack>,<unordered_map>,<unordered_set>,<vector>头文件不能被引用)。

二、数据格式

本次实验用到的数据结构如下所示:

id	name	Value1	Value2	Value3	…
int	string	int	int	int	int
主键	…	…	…	…	…

每条数据的第一列为主键id，第二列为一个字符串表示其name，后面有若干列，每列是一个int类型的数值。

实验要求可以从数据表文件导入数据库，文件格式为:

第一行为数据表名称;

第二行为列属性名描述，以空格隔开;

后面每一行为一条数据。

一个数据表文件的例子如下:

score

id	name	data_structure
0	xiaoming	99
1	xiaohong	100
2	xiaogang	98

……

……

……

三、功能要求

0、首先实现helper函数，用来解析命令，命令的具体格式见下面介绍。

1、增加数据

命令格式: INSERT (id, name, value1, value2, ...)

2、删除数据

命令格式: DELETE id //将编号为id的数据删除

DELETE id1 id2 //将编号$id \in [id1,id2]$的数据删除

3、修改数据

命令格式: SET id key value //将编号为id的数据属性key的值设置为value

ADD id key value //将编号为id的数据属性key的值增加value

SET id1 id2 key value //将编号$id \in [id1,id2]$的数据属性key的值设置为value

ADD id1 id2 key value //将编号$id \in [id1,id2]$的数据属性key的值增加value

4、查询数据

命令格式：QUERY id //查询编号为id的数据

QUERY name = str //查询name=str的数据

QUERY key > value //查询key属性大于value的数据，这里的'>'也可以是'<'、'='、'<='、'>='、'!='

QUERY id1 id2 key > value //查询编号$id \in [id1,id2]$区间内key属性大于value的数据，这里的'>'也可以是'<'、'='、'<='、'>='、'!='

QUERY key : k ASC//查询key属性的value第k大的数据，若为DESC则查询第k小的数据

QUERY id1 id2 key : k ASC//查询编号$id \in [id1,id2]$区间内key属性的value第k大的数据，若为DESC则查询第k小的数据

QUERY key : k LIST ASC//查询key属性的value前k大的数据，若为DESC则查询前k小的数据

QUERY id1 id2 key : k LIST ASC//查询编号$id \in [id1,id2]$区间内key属性的value前k大的数据，若为DESC则查询前k小的数据

SUM id1 id2 key //计算编号$id \in [id1,id2]$区间内key属性的value的和

上述操作保证只有一个数据表。

上述操作保证只有一个数据表。

5、集合操作：

命令格式：UNION table1 table2 //将两个表按name取并集，输出所有name，按先table1后table2的顺序

INTER table1 table2 //将两个表按name取交集，输出所有name，按table1的编号顺序

测试时从测试文件读取命令，每行一个命令，将结果写入结果文件，若一个操作有多个结果，按id升序排序输出，若操作失败，不输出。除了集合操作外，其他操作的结果均输出整条记录。

四、验收及评测

为了鼓励大家实现多种数据结构，我们将分组测试并对效率进行加权排名最后给出成绩

我们会准备多个测试文件，在代码里会指明这个测试文件属于哪一类，每一类保证只包含特定的操作，具体如下：

第一类：单点测试，包含单点增加、删除、修改和查询数据；

第二类：区间测试，数据保证在开始顺序插入完，后续无插入操作和删除操作；会出现区间修改，区间查询数据的所有操作；

第三类：除集合外所有操作；

第四类：插入数据，查询数据的前三种操作（单点查询），集合操作。

截止日期：2019.1.17日23点59分。

提交：将所有.h和.cpp文件打包成学号.zip提交，然后提交一份项目报告，word和pdf均可。

评测：我们会分四步测试，分别为上述四类测试。对每一种测试计时，在结果正确的基础上按效率单独排名，最终排名按各项测试排名加权和排序。我们对最低效率有所要求，请尽量使用高效的数据结构。