

# Project 3: NoSQL

---

Haoqi Wu

19212010008

## 1. Description

---

NoSQL(Not Only SQL), 意即"不仅仅是SQL"。在现代的计算系统上每天网络上都会产生庞大的数据量。这些数据有很大一部分是由关系数据库管理系统(RDMBS)来处理。通过应用实践证明, 关系模型是非常适合于客户服务器编程, 远远超出预期的利益, 今天它是结构化数据存储在网络和商务应用的主导技术。

NoSQL 用于超大规模数据的存储。(例如谷歌或Facebook 每天为他们的用户收集万亿比特的数据)。这些类型的数据存储不需要固定的模式, 无需多余操作就可以横向扩展。

今天我们可以通过第三方平台(如: Google,Facebook 等)可以很容易的访问和抓取数据。用户的个人信息, 社交网络, 地理位置, 用户生成的数据和用户操作日志已经成倍的增加。我们如果要对这些用户数据进行挖掘, 那SQL 数据库已经不适合这些应用了, NoSQL 数据库的发展也却能很好的处理这些大的数据。

在"NoSQL"这个词存在前, 像 memcached 这样的键/值数据存储(Key/Value Data Stores)无须table schema 也可提供数据存储的功能。实际上, 在K/V 存储时, 根本没有"表 (table)"的概念。只有键(keys)与值 (values)。

## 2. Preliminary

---

一个基于 Python 的最原始的 NoSQL 数据库, 简单来说就是一个大的字典 (dictionary)。它支持如下几种操作:

### PUT

- 参数: Key, Value
- 目的: 向数据库中插入一条新的条目 (entry)

### GET

- 参数: Key
- 目的: 从数据库中检索一个已存储的值

### INCREMENT

- 参数: Key
- 目的: 增长数据库中的一个整型值

## DELETE

- 参数: Key
- 目的: 从数据库中删除一个条目

## 3. Specification

---

2016 年度热搜榜显示,“郑爽过去一年119 次登上微博搜索榜, 42 次问鼎热搜榜第一位, 创造15 亿搜索热度, 霸屏138670 分钟。”这已经成为一种微博现象, 不仅能让一个明星保持高热度, 还能让一个八十线以外的普通人迅速为人所知。换句话说, 没有哪家媒体、哪个平台的议程设置能力可以超过微博, 微博扮演着一种现代的“广场”的角色, 而实时热搜几乎是微博上的舆论风向标, 在反映正在讨论的议题的同时, 更具备强烈的引导性。掌握了热搜, 就意味着掌握了舆论风向标, 获得直接的宣传效果, 也意味着中间可以存在各种变现空间。据《扬子晚报》报道称, 在一些电影的宣传费用中, 互联网投放占比超过四成, 其中微博营销占比很大, 话题、热搜次数已经成为一项必备的采购项目。有意思的是, “买榜”不仅包括让自己上榜, 相应地还能让一些词不上榜。“有明星买断屏蔽, 比如一个词, 明星不想要搜索出来, 他就可以买断, 搜这个词就没有结果。”

## Language: Java

### Question 1

需要你来做一个后台系统, 支持维护帖子阅读量。有以下几类操作:

1. 加入键值对 (键是字符串, 值是整数)
2. 给某个帖子增加阅读量, 即给某个键所对应的值加上某个值
3. 查询某个帖子的阅读量, 即返回某个键对应的值
4. 清空某个帖子的阅读量。即把某个键对应的值置0

### Question 2

维护人员认为日常操作中需要增加批量修改帖子阅读量的功能:

1. 给以某个字符串为前缀的每个帖子增加阅读量
2. 查询以某个字符串为前缀的帖子总阅读量
3. 清空以某个字符串为前缀的帖子的阅读量

### Question 3

负责这个项目的高管认为还应增加更高级的操作:

1. 给包含某个字符串的每个帖子增加阅读量
2. 查询包含某个字符串的帖子的阅读量
3. 清空包含某个字符串的帖子的阅读量

## Input Form

1. PUT Key, Value
2. ADD Key, Value
3. QUERY Key
4. DEL Key
5. ADDBEGINWITH Key, Value
6. QUERYBEGINWITH Key
7. DELBEGINWITH Key
8. ADDCONTAIN Key, Value
9. QUERYCONTAIN Key
10. DELCONTAIN Key

其中 `key` 为长度不超过 500 的字符串（只包含小写字母），`value` 为 长整形，在操作过程中，保证所有帖子的阅读量在 `long int` 范围。

## Output Form

顺序输出每条查询语句的值。

## Sample Input

```
PUT hellofdu, 1905
QUERY hellofdu
ADD hellofdu, 2017
QUERY hellofdu
PUT hellozhangjiang, 2005
QUERYBEGINWITH hello
PUT beautifulfdu, 2016
ADDCONTAIN fd, 2017
QUERYCONTAIN fd
```

## Sample Output

```
1905
3922
5927
9972
```

## Judge Criteria

每个数据点有不同时限，提交评测系统，通过即可获得相应分数。

第一组：只包含前4种操作，操作数量 $n(1 \leq n \leq 1000)$ ，分数10。

第二组：只包含前4种操作，操作数量 $n(1 \leq n \leq 100000)$ ，分数10。

第三组：只包含前7种操作，操作数量 $n(1 \leq n \leq 1000)$ ，分数10。

第四组：只包含前7种操作，操作数量 $n(1 \leq n \leq 100000)$ ，分数30。

第五组：包含各种操作，操作数量 $n(1 \leq n \leq 1000)$ ，分数10。

第六组：包含各种操作，操作数量 $n(1 \leq n \leq 100000)$ ，分数30。

# Submission

---

**Website:** This project is deployed in the **hustoj** and you need to **log in the website** (<http://10.141.222.205/>) **to fix the project and submit.**

- ***Please register / login use your student ID, or you will get ZERO points!***
- ***No Plagiarism or you will get ZERO points!***

**Deadline:** 26 Dec, 2019 18:00, any uploads after 26 Dec, 2019 18:00 wil get ZERO points.