**Redis的Key命名规范**

没有长度限制，但大小尽量不要超过1024字节

区分大小写

redis的key没有物理上的关联关系，但可以通过命名规范实现key的关联，例如：user:1:name 表名：id：属性名

redis推荐冒号分割数据

**String类型：**

单值缓存

set key value;

get key;

对象缓存

直接把对象json化存储

set user:1 value(json格式数据)

使用命名规范，将对象数据分开存储

mset user:1:name zhangliang user:1:password 1234

mget user:1:name user:1:password

推荐使用第二种方式，节省反序列化时间，方便对数据修改

分布式锁

利用redis单线程特性，使用setnx

setnx key value

del key

返回1代表取锁成功，返回0代表取锁失败

在业务执行完毕后通过删除数据释放锁

set key value ex 10 nx

设置带有过期时间的锁，防止线程以外终止导致死锁

计数器

通过incr命令，记录文章的访问量，

例：incr article:readcount:{文章id}

get article:readcount:{文章id}

web集群的session共享

spring session + redis实现session共享

分布式全局序列号

**Hash类型：**

对象缓存

hmset user {userID}:name zhanghua {userID}:password 1234

通过hmset存储对象对应字段的值

电商购物车

购物车添加商品

hset cart:{userID} {商品id} {购买个数} ...

通过这种格式将实现购物车。

hset cart:1001 1008101 1

增加购买个数

hincrby cart:1001 1008101 1

删除商品

hdel cart:1001 1008101

获取购物车内的所有商品

hgetall cart:1001

redis中不宜存储过大数据，商品名称等信息通过id异步加载到页面

hash与string的对比：

哈希数据归类整合储存，方便管理

相比string消耗内存和cpu更小

节省空间

过期时间不能设置在field上，只能用在key上

redis集群架构下不适合大规模使用

**List类型**

list实现栈

lpush + lpop 栈(Stack)

list实现队列

lpush + rpop 队列(Queue)

list实现阻塞队列

lpush + brpop 阻塞队列(Blocking MQ)

brpop 键 超时时间：从list中取值，如果list中不存在值，则进入阻塞，直到取到值或者时间超时

blpop 原理同上

微信微博消息流模型

push推送信息模型：

lpush msg:1888 10006

lrange msg:1888 0 5

将新发布的文章的id信息lpush到关注用户消息列表里，当用户lrange时就可以获取到最新消息

pop拉取消息模型：

适用于有粉丝用户较多的大微博

用户在大微博链表中拉取消息，并将各大微博的消息进行时间排序

**Set类型**

微信抽奖小程序

将参与抽奖的用户放入到set集合里

sadd {key} {用户id}

查询所有参与抽奖的用户名单

smembers {key}

从参与名单中抽取固定人数中奖

srandmember {key} {奖励人数}

spop {key} {奖励人数}

微博点赞功能实现

将点赞用户的id放入文章的set集合里面

sad like:{消息id} {用户id}

取消点赞

srem like:{消息id} {用户id}

检查用户是否已经点过赞

sismember like:{消息id} {用户id}

获取点赞用户的列表

smembers like:{消息id}

获取点赞用户数量

scard like:{消息id}

微信微博的关注模型

利用Set的数学集合api

交集集合实现：共同关注的人

使用sismember实现：我关注的人也关注了他

利用差集集合实现：我可能认识的人