5种数据类型 string、list、set、zset、hash

1：查看所有键值对

keys \*

2：判断key是否存在

exists key

3：查看key对应valueleix

type key

4：删除key

del key 直接删  
unlink key 非阻塞式io删

5：设置key过期时间

expire key 10 10S过期

6：查看key过期时间

ttl key -1 不过期，-2已经过期

7: 切换数据库 默认15个库 0-15，默认0号库

select 1

8：查看当前库有多少key

dbsize

9：清空当前库

flushdb

10 字符串类型的value最多512M

11 将数字value 加1减1

incr key

decr key

12 将数字value 加n减n

incrby key n

decrby key n

13 设置多个key vlue

mset key1 v1 key2 v2

msetnx k1 v1 k2 v2 当k1,k2都不存在时候才能设置成功

14 主从模式

slaveof 主IP port 设置成某个的从服务器

一主二仆：配合哨兵模式

星火相传：链式结构 A 主 B, B主C

反客为主：需要人为干预

15查看redis状态

info replication

16 设置时候加锁

set user 10 nx ex 5

设置key user ，value 10，nx表示上锁，ex表示失效时间5秒

缓存穿透：大量请求过来，缓存中查询不到数据，直接查询数据库，然后导致奔溃

产生原因：1redis中查询不到数据了，2操作中存在很多非正常访问

解决办法：

1：对空值进行缓存

2：设置访问白名单

使用bitmap类型定义一个可以访问的白名单，名单id作为偏移量，每次访问和id对比

3：采用布隆过滤器（第二种方案优化，原理一样）

4：进行实时监控

缓存击穿：

现象：1数据库访问压力瞬时增大

2redis里面没有出现大量key过期

3 redis正常运行

产生原因：

1：redis某个key过期，但是大量的访问使用这个key

解决办法：

1：预先设置热门数据

2：实时调整

3：使用锁

缓存雪崩：

现象：

1：数据库压力变大，服务器崩溃

产生原因：

1：短时间内大量key过期

解决办法：

1：构建多级缓存架构

2：使用锁或队列 （访问数据库加锁）

3：设置过期标志，更新缓存

4：将缓存失效时间分散开

分布式锁

1：基于数据库实现

2：基于redis实现

3：基于zookeeper