命令

Del key

Exist key

Type key

Expire key

**字符串(string)**

Set key value

Mset key1 value1 key2 value2 …

Mget key1 key2…

Setnx key value(set if key not exist)

Incr key/decr key

Incrby key increment/decrby key decrement

Strlen key

Append key value

**散列(hash)**

Hset key field value

Hmset key field1 value1 field2 value2….

HSETNX key field value(set if filed not exist)

Hget key field

Hmget key filed1 field2

Hgetall key

Hexist key field

Hdel key field

HINCRBY key field increment

Hlen key(field count)

**列表(list)**

Lpush key value1 value2…

Rpush key value1 value2…

Lpop key

Rpop key

Llen key

Lrange key start stop

Lindex key index

Lset key index value

Ltrim key start end (删除[start,end]之外元素)

LREM key count value 删除列表中前count个值为value的元素；

**集合(Set)**

Sadd key member1 member2…

Srem key member1 member2..

Smembers key…

Sismember key member

Scard key(return count)

Sinter key1 key2…

Sunion key1 key2 …

SPOP key(随机弹出)

**有序集合(zset)**

ZADD key score1 member1 [score2 member2 ...]

ZSCORE key member

Zrange key start stop [withscores]

Zrevrange key start stop [withscores]

Zrangebycore key min max [withscores] [LIMIT offset count]

ZINCRBY key increment member

ZCARD key

ZREM key member1 [member2 …]

Zremrangebyrank key start stop

Zremrangebyscore key min max

(交集并集)

**事物**

redis事务能够保证一个事务内的命令依次执行而不会在中途插入其它命令；因为redis本身是单线程处理，所以相当于序列化的事务隔离级别；

事物

MULTI

描述：标识事务开始的命令，后面对数据操作的命令将会被放入队列，暂不执行；

返回值：OK

EXEC

描述：执行暂存队列里面的命令，相当于commit命令；

返回值：事务队列中命令执行返回值列表；

Watch

命令用于监视一个(或多个) key ，如果在事务执行之前这个(或这些) key 被其他命令所改动，那么事务将被打断（muLti之前）

SORT key [ALPHA] [LIMIT offset count] [BY key | key->字段名] [ASC | DESC] [GET key | key->字段名]+ [STORE destkey]

说明：

SORT命令可以对列表类型、集合类型、有序集合类型键进行排序，并且可以完成与关系数据库中的连接查询相类似的任务；

.Redis支持两种持久化方式：

1）RDB方式：根据指定规则，定时将内存中的数据存储在硬盘上；类比Oracle数据库的exp导出方式；

2）AOF方式：将每次执行的命令记录下来；类比Oracle数据库的SQL导出方式；

RDB（copy-on-writer）

1）使用fork函数，复制当前进程的副本，作为子进程；复制过程采用copy-on-writer策略；

2）父进程继续处理客户端发来的命令，子进程负责将内存中数据写入硬盘中的临时文件；

3）当子进程写入完所有数据后，会用该临时文件替换旧的RDB文件，完成一次快照；

AOF

1）将redis执行的每一条更改数据的命令都追加到硬盘文件中；

2）由于每次执行命令都有写入硬盘操作，会降低redis性能；

3）相比RDB方式，更能够保证数据的一致性；

由于操作系统缓存机制，每次执行的写命令数据并没有真正写入硬盘，而是进入了操作系统的硬盘缓存，系统默认30s执行一次同步操作

**优化**

1）精简键名和键值，但保持可读可用；  
2）优化内部编码方式；