**1.select 0-15：**切换redis库；

**2.keys \*：**查看所有key；

**3. FLUSHDB：**清空数据库;

**4.FLUSHALL：**清空所有数据库；

**5.move k1 2：**将k1移动到2号库；

**6.事务执行步骤：**

四种结果：正常执行，放弃事务，全体连坐(语法有问题时非运行时报错，都不执行)，冤头债主(语法没有问题时运行时报错，只是报错的不执行)，watch监控；

①MULTI：开启事务；

②添加操作加入队列，如set k1 v1;

③EXEC：入栈；

④DISCARD：放弃事务；

⑤WATCH key：监视key，如果在事务执行之前这个key被其他命令所改动，那么事务将被打断，类似乐观锁；

⑥UNWATCH：取消watch命令对所有key的监视；

**7.一主多从，主从复制，读写分离**

①info replication：查看主机或是从机；

②slaveof 127.0.0.1 6379：设置从机跟随主机，只要连接成功，主从复制就开启；

③slaveof no one：使当前数据库停止与其他数据库同步，转为主数据库；

注意：

加入从机，从机也会将连接主机之前对主机操作的数据复制下来，全量复制，随后增量复制；

主机宕机，从机还是从机，等主机恢复自动连接主机，一样主从复制；

从机宕机，等从机恢复，将变成主机，可进行配置，恢复还是从机；

**8.哨兵模式：**

①创建sentinel.conf配置文件；

②sentinel monitor host主机名 127.0.0.1 6379 1；监控的主数据库，从机竞争多余1票选为主机；

注意：

主机宕机，从机之一选为主机，主机恢复，将自动配置为从机；

**9.redis五大类型：**

**String**：key和value都是string类型；

​方法：①set/get：存值/取值；

expire k1 10: 为k1设置过期时间10秒，自动删除k1，ttl k1：查看剩余过期时间；

persist k1：删除过期时间，设置k1永不失效，与expire相对应；

exists k1: 查看k1是否存在；

②append：在value后追加内容；

strlen k1：返回k1的长度；

③incr/decr：在value为数字时，数字加或减1；如：incr k1；

incrby k1 10:递增10；decrby k1 10:递减10；

incrbyfloat k1 0.1：k1小数递增0.1；

④setex/setnx/del：setex是设置过期时间，如：setex k1 10 v1，k1的过期时间为10秒后；

setnx job "hello" : 设置分布式锁

如果当前job存在，则返回0，表明赋值不成功。

如果当前job不存在，则返回1，表明赋值成功。

del job: 单独key及关联所有value，适用String，List，Hash，Set ，Zset

⑤getrange/setrange：getrange是截取value个数，如：

getrange k1 0 3，把k1的value截取0-3个字符， getrange 0 -1:查询所有的，为-2时取n-1字符；

setrange是替换字符串，如：setrange k1 0 xxx，把k1的value从0位开始将前三位替换为xxx；

⑥getset：查询出旧值后再设置新值，getset k1 lucong：先查询返回k1的值，再设置k1新值；

⑦mset/mget/msetnx：mset可以一次性插入多个key和value，mset k1 v1 k2 v2；

mget可以一次性获取多个key的值；

msetnx可以一次性插入多个，首先先做不存在判断，部分存在也是不能插入库中；

**List**：使用的LinkedList，双向链表形式，前后都可以插入数据；

方法: ①lpush/rpush/lrange：lpush是\*\*先进后出\*\*，如：lpush list01 1 2 3 4 5，展示为5 4 3 2 1；

rpush:是\*\*先进先出\*\*，如：rpush list02 1 2 3 4 5，展示为1 2 3 4 5；

lrange:查看范围，如：lrange list01 0 -1，0到-1范围区间是查看所有；

②lpop/rpop：从缓存中消失；

lpop左出站，如：lpop list01，弹出5；lpop list02，弹出1；

rpop右出站，如：rpop list01，弹出1；rpop list02，弹出5；

③lrem list01 count value：删除集合中n个元素，如：

count = 0：lrem list01 0 3，删除集合中所有为3的value；

count > 0：lrem list01 2 3，排序后从表头到表尾删除集合中2个为3的value；

count < 0：lrem list01 -2 3，排序后从表尾到表头删除集合中2个为3的value;

④lindex：查看角标索引值，如：lindex list01 2，查看list集合角标索引为2的数据；

⑤lset：将指定位置的数据改值，如lset list01 1 x，将list01的角标索引为1的值改为x；

⑥ltrim：截取指定长度数据，如：ltrim list01 0 3；截取list01集合0-3长度数据，替换list01数据；

⑦llen：查看list长度，如：llen list01；

⑧linsert：将值插入value之前或之后：

如linsert list01 before/after x java，将java插入集合x值的之前或之后；

⑨rpoplpush：将压栈放入另一个栈顶，如rpoplpush list01 list02，将list01的压栈放入list02集合栈顶，并删除list01压栈元素；

**Hash**：相当于Map<String, Object>，KV模式保持不变，但V是个键值对；

方法:①hset/hget：插入，获取，如hset user id 12，hget user id;

hmset/hmget：一次性插入多条记录，如hmset user id 13 name z3 age 17，hmget user id name age;

hgetall：获取所有的hash键值对，hgetall user;

hdel：删除hash键值对，hdel user name;

②hexists：判断是否存在key，hexists map01 id，查询map01中是否存在key为id；

③hsetnx：判断不存在就插入，存在不做操作，如hsetnx user email 123@qq.com;

④hincrby/hincrbyfloat：value为数字或者小数时进行递加，如hincrby user age 3， hincrbyfloat user score 0.5;

⑤hkeys/hvals：获取map中所有的key或value，如hkeys user，hvals user;

⑥hlen：获取KV数量，hlen user;

**Set**：无序无重复；

方法:①sadd：添加set集合内容，如sadd set01 2 2 3 3 ;

smembers：查看set集合所有内容， 如smembers set01;

sismember：查看值是否存在set集合中，如sismember set01 2;

②scard：查看集合中有多少个元素，如scard set01;

③srem：删除集合中的元素，如srem set01 2;

④srandmember：随机抽取集合中指定个数元素，如：

srandmember set01 2，set01集合中随机取两个元素，不写count随机获取一个；

⑤spop：随机 出栈，如spop set01 1，随机出栈一个，从内存中删除；

⑥smove：从一个set集合中数据移动另一个set集合，如：

smove set01 set02 5；将set01集合中的5移动到set02中，不存在set02就会创建，然后添加元素；

⑦sdiff/sinter/sunion：sdiff，差集，在第一集合中而不再第二个集合中，如sdiff set01 set02，在set01集合中而不在set02中；sinter，交集；sunion，并集；

**Sorted Set**：有序无重复，value为键值对；

方法: ①zadd 集合 score value：添加set集合，如：

zadd zset01 60 v1 70 v2 80 v3，如果val值相等但score不同，val唯一，score最后一次修改；

zrange：正序输出所有，如zrange zset01 0 -1，查看所有的value，zrange zset01 0 -1 withscores，按照score升序，查看所有的key和value；

zrevrange：逆序输出所有，如zrevrange zset01 0 -1;zrevrange zset01 0 -1 withscores，按照score升序，查看所有的key和value；

②zrangebyscore：指定范围截取，如：

zrangebyscore zset01 60 70，zrangebyscore zset01 (60 (80，不包含60和80；

zrangebyscore zset01 60 80 limit 2 1，包含60和80从索引下标为2截取1个元素;

zrangebyscore zset01 -inf +inf limit 0 2: 分页正序输出前两条数据；

③zrevrangebyscore：倒序输出数据， 如：

zrevrangebyscore zset01 80 60；

zrevrangebyscore zset01 +inf -inf limit 0 2 :分页倒序输出前两条记录。

④zrem：删除set集合元素，如zrem zset01 v3;

⑤zcard/zcount：统计个数，如zcard zset01，zcount zset01 60 80，范围统计60到80的个数；

⑥zscore/zrank/zrevrank：获取score排序值/正序获取索引下标/逆序获取索引下标，

如zscore zset01 v4，zrank zset01 v4，zrevrank zset01 v4;

**10. 扩展知识：**

①redis每秒读11万次，每秒写8万次；

②持久化方式：

RDB:能够在指定的时间间隔能对数据进行快照存储；

AOF:记录每次对服务器写的操作，当服务器重启的时候会重新执行这些命令来回复原始的数据，AOF命令以redis协议追加保存每次写的操作到文件末尾，Redis还能对AOF文件进行后台重写，使得AOF文件的体积不至于过大，和RDB同时开启持久化，先加载AOF文件；

③主从复制：主机可读可写，从机只提供读服务；首次加入从机会全量复制主机内容；当主机down掉，从机还是从机死等；主机down掉后回归，主从复制从新开启；从机down掉回归，变为主机，如果配置文件配置slaveof <masterip> <masterport>：回归继续为从机；主机down掉，使用哨兵投票模式，可以将从机设置为主机；

缺点：由于所有的写操作都是先在master上操作，然后同步更新到slave上，所以从master同步到slave机器有一定的延迟；

④info replication：查看redis状态信息，配置参数；

⑤slaveof 127.0.0.1 6379：主从复制，配从不配主，从机挂载的主机地址；

⑥slaveof no one：反客为主，主机down掉，设置从机为主机

⑦缓存穿透：是指查询数据库一定不存在的数据。解决：如果从数据库查询的对象为空，也放入缓存，只是设定的缓存过期时间较短，比如设置为60秒。

缓存雪崩：是指在某一个时间段，缓存集中过期失效。解决：一般是采取不同热门数据，缓存不同周期。

缓存击穿：是指一个key非常热点，在不停的扛着大并发，大并发集中对这一个点进行访问，当这个key在失效的瞬间，持续的大并发就穿破缓存，直接请求数据库，就像在一个屏障上凿开了一个洞。解决：热门数据设置缓存永不过期。

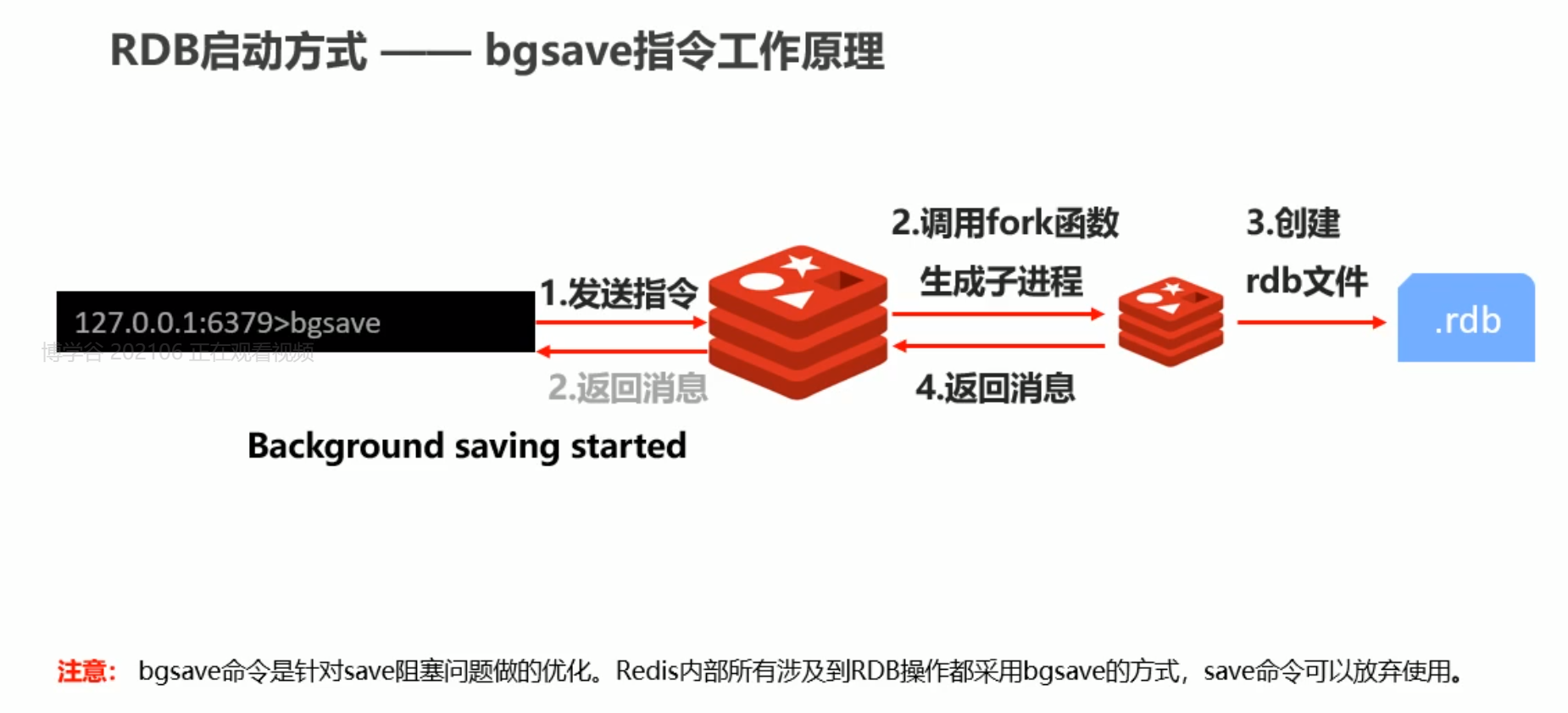
**11. 深度理解持久化：**

RDB:

①了解save命令工作原理：



②为解决save指令阻塞问题，采用bgsave(后台处理保存)模式，也是RDB正在使用的方式，工作原理：



​ ③save配置模式：**save** second changes：设置自动持久化的条件，满足限定时间范围内key的变化数量达到指定数量即进行持久化，底层就是使用的bgsave模式。

④RDB优/缺点：

