**Redis笔记**

**【第一节】**

1. **什么是NOSQL？**

**NOSQL：即Not-Only SQL(泛指关系型的数据库)，作为关系型数据库的补充。**

1. **NOSQL的作用**

**应对基于海量用户和海量数据提前下的数据处理问题**

1. **NOSQL的特征：**
2. **可扩展，可伸缩**
3. **大数据量下高性能**
4. **灵活的数据模型**
5. **高可见**

**【第二节】**

1. **什么是redis？**

**Redis(Remote Dictionary Server)是用C语言开发的一个开源的高性能的键值对(key-value)数据库。**

1. **redis的特征：**
2. **数据间没有必要的关联关系**
3. **内部采用单线程机制进行工作**
4. **高性能**
5. **多数据类型支持：string list hash set sorted\_set**
6. **持久化支持。可以进行数据灾难恢复**
7. **redis的应用**
8. **热点数据**
9. **任务队列**
10. **即时信息查询**
11. **时效性信息控制**
12. **分布式的数据共享**
13. **消息队列**
14. **分布式锁**

**【第三节】**

1. **功能性命令**
2. **信息添加**
3. **功能：设置key，value数据**
4. **命令：set key value**
5. **信息查询**
6. **功能：根据key查询对应的value，如果不存在，返回空(nil)**
7. **命令：get key**
8. **清除屏幕信息**
9. **清除屏幕信息**
10. **功能：清除屏幕中的信息**
11. **命令：clear**
12. **帮助信息查阅**
13. **帮助**
14. **功能：获取命令帮助文档，获取组中所有命令信息名称**
15. **命令：help 命令名称**

**Help @组名 例：help @string**

1. **退出指令**
2. **退出**
3. **功能：退出客户端**
4. **命令：quit、exit、<ESC>按键**

**【第四节】**

**十一、业务数据的特殊性**

1. **做缓存使用**
2. **原始业务功能设计：用于秒杀、618活动、双11活动、排队购票等**
3. **运营平台监控到的突发高频访问数据：用于突发时政要闻、被强势关注围观**
4. **高频、复杂的统计数据：用于在线人数、投票排行**
5. **附加功能**
6. **单服务器升级集群**
7. **Session管理**
8. **Token管理**

**【第五节】**

**十二、redis数据存储格式**

1. **Redis自身是一个map，其中所有的数据都是采用key:value的形式存储**
2. **数据类型指的是存储的数据的类型，也就是value部分的类型，key部分永远都是字符串**

**十三、string类型**

1. **存储的数据：单个数据，最简单的数据存储类型，也是最常用的数据存储类型**
2. **存储数据的格式：一个存储空间保存一个数据**
3. **存储内容：通常使用字符串，如果字符串以整数的形式展示，可以作为数字操作使用**
4. **String类型数据的基本操作**
5. **添加/修改数据：set key value**
6. **获取数据：get key**
7. **删除数据：del key**
8. **添加/修改多个数据：mset key1 value1 key2 value2**
9. **获取多个数据：mget key1 key2 …**
10. **获取数据字符个数(字符串长度)：strlen key**
11. **追加信息到原始信息后部(如果原始信息存在就追加，否则新建)：append key value**
12. **单数据操作与多数据操作：单数据要比多数据操作慢**

**009**