1. 发布订阅
2. 命令

发布：PUBLISH <channel> <message>

订阅频道：SUBSCRIBE

订阅模式：PSUBSCRIBE

1. 实现

**频道：**redis将所有频道的订阅关系都保存在服务器状态的pubsub\_channels字典里，这个字典的键是被订阅的频道，而键的值则是一个链表，链表里面记录了所有订阅这个频道的客户端。执行订阅频道时，如果频道已经有其他订阅者，那么它在pubsub\_channels字典中必然有相应的订阅者链表，redis就将当前的客户端添加到订阅者链表的末尾；如果频道还未有任何订阅者，那么这个pubsub\_channels字典中肯定没有这个频道，redis首先会在pubsub\_channels字典中为该频道创建一个键，并将这个键设置为空链表，然后再将客户端添加到链表。退订与订阅的逻辑相反，是在字典中找到频道对应的订阅者链表，从这个链表中删除该客户端的结点信息。

**模式：**redis将所有模式的订阅关系都保存在服务器状态的pubsub\_patterns属性里。pubsub\_patterns属性是一个链表，链表中的每个节点都包含一个pubsubPattern结构体，这个结构体有两个属性，pattern属性记录了被订阅的模式，client属性则记录了订阅模式的客户端。执行订阅模式时，首先将新建一个pubsubPattern结点，将结点的pattern属性设置为被订阅的模式，client属性设置为订阅模式的客户端；将pubsubPattern结点添加到pubsub\_patterns链表的尾部。

**发布：**发布过程分两步，将message消息发送给channel频道的所有订阅者，将message消息发送给pattern模式的订阅者。第一步在pubsub\_channels字典里找到频道channel的订阅者客户端，将消息发送给这些客户端；第二步遍历整个pubsub\_patterns链表，查找那些与channels频道相匹配的模式，然后将消息发送给这些客户端。

1. 事务
2. 事务的过程

事务首先以一个MULTI命令为开始，接着将多个命令放入事务队列中，最后由EXEC命令将事务提交给服务器执行。

1. 事务的实现

每个客户端都有自己的事务状态，这个事务状态保存在客户端状态的mstate属性中。这个事务状态的类型为multiState，multiState包含一个事务队列\*commands和一个计数器count，事务队列中的每个元素都保存了命令指针，命令参数，参数数量。执行EXEC命令会将事务队列中的命令按顺序取出执行。

1. WATCH

WATCH命令是一个乐观锁，一般使用在EXEC命令之前，监视任意数量的数据库键。在EXEC命令执行时，检查被监视的键是否有被修改过的。如果有，服务器拒绝执行事务。

1. WATCH实现

redis保存着一个watched\_keys字典，字典的键是被WATCH命令监视的数据库键，字典的值是一个链表，链表中是所有监视该键的客户端。所有对数据库进行修改的命令，在执行之后会调用touchWatchKey对watched\_keys字典进行检查，是否有客户端正在监视的键被修改。如果有的话，会将该客户端的REDIS\_DIRTY\_CAS标识打开，表示该客户端的事务安全性已经被破坏。服务器在收到客户端收到的EXEC命令时，服务器会根据这个客户端是否打开了REDIS\_DIRTY\_CAS标识来决定是否执行事务。

1. Redis事务的ACID

原子性：Redis不支持事务回滚机制，即使事务队列中的某个命令在执行期间出现了错误，整个事务也会继续执行。

隔离性：Redis事务总是以串行的方式运行的，所以Redis的事务是有隔离性的。