# Redis数据库简介

## Redis数据库的原理

1. Redis数据库保存在Redis服务器状态server.h/redisServer结构的db数组中。这个数组中的每个元素都是一个server.h/redisDb结构，而每一个redisDb结构就代表一个Redis数据库。
2. Redis服务器状态有一个dbnum属性，该属性用于在启动Redis服务器时，决定创建多少个数据库。Redis配置文件中的databases选项决定了dbnum属性的值。在默认情况下dbnum属性值为16，也就是在启动Redis服务器时，会默认创建16（0~15号）个数据库。
3. 客户端连接服务器后，默认使用的数据库是0号数据库，如果需要操作其他数据库，则可以使用select命令来切换数据库。
4. 在Redis服务器内部，有一个db属性，这个属性是客户端状态redisClient中的属性，它是一个指向redisDb结构的指针，用于记录当前客户端操作的目标数据库。redisClient.db指针指向redisServer.db数据中的某个元素，这个被指向的元素就是客户端的目标数据库。客户端连接服务器后，db属性默认为0，所以这个客户端默认使用的数据库也就是0号数据库。
5. select命令切换数据库的原理：使用select命令的index参数来切换到要操作的数据库，这个index就相当于redisServer.db数组的下标，这个下标从0开始，它也是redisClient.db指针的索引值，索引值从0开始。通过指定index的值，也就是修改redisClient.db指针，来指向服务器中的不同数据库，达到切换数据库的目的。

## 数据库的切换

客户端连接上服务器后，默认使用的0号数据库。如果要想操作其他的数据库，则可以使用select语句进行切换。命令如下：

**select index**

index表示要切换的数据库的索引，从0开始。

# Redis数据库中的键操作

1. Redis是一个键值对数据库，在Redis服务器中使用server.h/redisDb结构来表示Redis数据库。
2. 在redisDb结构中存储了Redis数据库中的所有键值对，它就是dict字典，这个字典就是Redis的键空间，这个键空间存储了Redis数据库中的键和其对应的值，每个键就是一个字符串对象，而每个键对应的值可以字符串对象、哈希对象、列表对象、集合对象、有序集合对象。
3. 在数据库中执行键的操作时，如添加键、修改键、删除键等，其实就是在操作Redis的键空间。

## 添加键

根据要设置的值的类型不同，添加键的命令也不同，可参考第3章。

## 取键值

根据值的类型不同，获取值的命令也不同，可参考第3章。

## 修改键的值

根据值的类型不同，修改值的命令也不同，可参考第3章。

## 删除键

不管值是什么数据类型的，当一个键不再使用时，都是使用del命令对其进行删除，可参考第4章，键的删除命令。

在实际应用中，针对过期的键，我们也需要删除。删除过期键有3种方式：

1. 定时删除

为一个键设置生存时间的同时，为该键设置一个定时器（Timer），当这个键过期时，这个定时器会立即执行对键的删除操作。这种删除过期键的方式比较友好，利用定时器定时删除键可以保证过期的键不会长时间保留在数据库中，从而释放空间。但这种方式也有缺点：它会占用CPU的大量时间，如果过期的键比较多，同时CPU又比较繁忙，则会对服务器的吞吐量和响应时间产生很大影响，从而影响服务器的性能。

1. 惰性删除

对数据库中的过期的键持放任态度，但是每次从键空间中获取键时，都会对获取的键进行检查，判断它是否过期，过期就删除它，没有过期就返回它。惰性方式虽然不消耗CPU时间，但是它不会定时定期的删除过期的键，因此这些过期的键就会长期存留在内存中，占用大量的内存，这是惰性删除的最大缺点。

1. 定期删除

采用算法实现每隔一段时间就对数据库进行一次检查，筛选出过期的键进而删除。在算法中可以设置检查多少个数据库，每检查一次具体删除多少过期的键。定期删除方式是定时删除与惰性删除的折中方案。

其实，Redis服务器只采用了惰性删除和定期删除两种方式，这两种方式的实现如下：

·惰性删除的实现：Redis的db.c/expireIfNeeded函数实现了对过期键的惰性删除。Redis的所有读/写命令在读/写数据之前都会调用expireIfNeeded函数对输入键进行检查。常见的读/写命令有set、sadd、lrange、hget、keys等。

·定期删除的实现：Redis的server.c/activeExpireCycle函数实现了对过期键的定期删除。前面说过，定期删除过期的键时采用算法实现的，算法会设置周期时间，每当Redis服务器周期性操作server.c/serverCron函数时，它会调用activeExpireCycyle函数在规定的时间内开始遍历各个数据库，从数据库的过期（expires）字典中随机筛选出部分过期的键，随后删除。

# Redis数据库通知

数据库通知功能实现了让客户端通过订阅指定的消息频道或信息模式，来动态获取数据库键的变化，以及数据库中命令的执行情况。

## 数据库通知分类

数据库通知分为键空间通知和键事件通知。

1. 键空间通知主要关注某个键执行了什么命令。

**subscribe \_ \_keyspace@<db>\_ \_: key**

key表示键名；

<db>表示几号数据库，如0表示0号数据库；

1. 键事件通知主要关注某个命令被什么键执行了。

**subscribe \_ \_keyevent@<db>\_ \_:set**

set表示set命令，你也可以指定为其他的命令，如get等。

<db>表示几号数据库，如0表示0号数据库；

1. 键空间通知使得客户端可以通过订阅频道来接收那些以某种方式修改了Redis数据集的事件，通过Redis的消息订阅发布功能实现事件的分发。因此，只要是支持消息订阅发布功能的客户端，都可以在不修改任何内容的情况下使用键空间通知功能。
2. Redis的消息订阅发布功能采取的是发送即忘的策略。当订阅事件的客户端断线时，所有在线期间分发给它的事件都会丢失。
3. 键空间通知针对每个修改数据库的操作，都会发送两种不同类型的事件。比如在1号数据库中执行了set name命令，系统会发送两条信息，相当于执行了publish命令：

publish \_ \_keyspace@1\_ \_:name set

publish \_ \_keyevent@1\_ \_:set name

频道\_ \_keyspace@1\_ \_:name可以接收1号数据库中所有修改name键的事件，而频道\_ \_keyevent@1\_ \_:set则可以接收1号数据库中所有执行set命令的键。

键空间通知以keyspace为前缀，键事件通知以keyevent为前缀。

在执行set name命令时：

·键空间频道的订阅者将接收到被执行的事件的名字，这里为set；

·键事件频道的订阅者将接收到被执行事件的键的名字，这里为name；

## 开启数据库通知

在默认情况下，数据库通知功能是关闭的，因为它会占用CPU资源。

在实际应用中，如果需要开启或关闭数据库通知功能，则可以通过config set命令来完成，或者修改Redis的配置文件redis.conf中notify-keyspace-events参数的值。

notify-keyspace-events参数的值可以是以下字符的任意组合，它们规定了服务器会发送哪些类型的通知。（db表示几号数据库，格式如\_ \_keyspace@1）

·K：表示键空间通知，所有键空间通知都以\_ \_keyspace@<db>\_ \_为前缀；

·E：表示键事件通知，所有键事件通知都以\_ \_keyevent@<db> \_ \_为前缀；

·g：表示del、expire、rename等与类型无关的通知命令的通知；

·$：表示字符串命令的通知；

·h：表示哈希命令的通知；

·l：表示列表命令的通知；

·z：表示有序集合命令的通知；

·s：表示集合命令的通知；

·x：表示过期事件，当有过期的键被删除时触发发送通知；

·e：表示驱逐事件，当有键因为maxmemory政策而被删除时触发发送通知；

·A：表示g$hlzsxe的别名；

但需要注意的是，在为notify-keyspace-events参数设置值时，至少要有一个K或E，否则无论其余的参数值是什么，都不会发送任何通知。

常见的参数值组合形式有以下几种：

·AKE：表示所有类型的通知都会被发送；

·AK：表示让服务器发送所有类型的键空间通知；

·AE：表示让服务器发送所有类型的键事件通知；

·K$：表示让服务器只发送与字符串类型键相关的键空间通知。其他类型的也是如此，如Kh表示只发送与哈希类型键相关的键空间通知；

·El：表示让服务器只发送与列表类型键有关的键事件通知，其他类型的也是如此，如Eh表示只发送与哈希类型键有关的键事件通知。

## 数据库通知的实现原理

数据库通知的发送功能是由notify.c/notifyKeyspaceEvent函数实现的，这个函数的定义如下：

void notifyKeyspaceEvent (int type , char \*event , robj \*key , int dbid ) ;

type参数：表示要发送的通知类型，程序会按照这个值来判断通知是否就是服务器配置的notify-keyspace-events参数所设定的通知类型，然后决定是否发送通知。

event参数：表示发送事件的名称；

key参数：表示产生事件的键，也就是与该发送事件相关的键；

dbid参数：表示数据库ID，也就是产生事件的数据库编号。

1. 函数notifyKeyspaceEvent会根据它的参数来生成事件通知的内容，以及接收事件通知的频道名。
2. 当一个Redis命令需要发送数据库通知的时候，该命令的实现函数就会调用notifyKeyspaceEvent函数，来实现发送通知功能。
3. 当Redis命令对数据库进行修改之后，服务器就会根据相关配置文件的设置来向客户端发送数据库通知。