



密集	内部公开
----	------

鸟域道场

产品研发 Redis 数据库设计规范

(V1.0)

Nander 研发团队
2026 年 6 月



文档信息

文档编号				发布版本	
起草时间	2026-06-08	定稿时间		当前版本	V1.0
起草人	姓名	部门	电话	电子邮件	
	商鞅	鸟域行政司		26089183@qq.com	
审阅人			签发人		
文档修改记录					
序号	修改时间	修改人	主要修改内容		存档版本
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					



目录

1 目的	4
2 适用范围	4
3 数据库设计规范	4
3.1 基础规范	4
3.2 键规范	4
3.3 值规范	5
3.4 数据类型选用规范	5
3.5 过期时间设计规范	6
4 缓存策略	6
5 淘汰策略	6
6 命令使用规范	6
6.1 强制禁用/限制命令	6
6.2 常规命令规范	7
6.3 客户端连接规范	7
6.4 运维与安全规范	7
7 工具	8
7.1 Redis Insight	8
8 附则	8
9 参考文献	8



1 目的

为了提升研发、测试、运维等技术人员基于Redis做出适合线上业务的缓存设计，需要规范Key命名、数据存储、命令调用、缓存策略等行为，规避缓存穿透、缓存击穿、缓存雪崩、内存溢出、命令滥用等线上问题，保障业务系统稳定、高效运行，特制定本规范。

2 适用范围

本规范适用于组织内所有自研产品与交付类项目的基于Redis的缓存、分布式锁、限流、计数器等业务场景的设计。

3 数据库设计规范

3.1 基础规范

- 1、【强制】Redis仅用作缓存中间件，禁止替代关系型数据库做持久化主存储。核心业务数据、强事务数据必须落地数据库，Redis仅做热点数据加速。
- 2、【强制】线上Redis禁止存储永久有效、永不淘汰的冷数据，全量业务缓存必须配置过期时间。
- 3、【强制】大数据文件（图片、视频、文档、超大文本）禁止直接存入Redis，仅存储文件URL/地址，特殊场景需存储二进制大文件必须提交审批。
- 4、【强制】生产环境必须使用集群/主从 + 哨兵架构，禁止单机Redis直接上线，保障高可用与故障自动切换。
- 5、【强制】Redis不承担复杂计算逻辑，统一交由应用程序处理，降低Redis服务压力。
- 6、【建议】读写分离：读请求优先走从节点，写请求统一落到主节点；核心高并发业务严格执行读写分离。

3.2 键规范

- 1、【强制】统一使用小写字母、数字、冒号组合，禁止大写、中文、特殊字符、空格；禁止拼音与英文混用，统一分层命名格式做到见名知意，比如：业务模块:业务标识:功能名称。
- 2、【建议】保证语义的前提下控制key的长度，过长Key会增加内存开销、降低查询效率，比如：user:{uid}:friends:messages:{mid}简化为u:{uid}:fr:m:{mid}。
- 3、【建议】集群模式下合理规划槽位，将Key中变化部分前置，防止出现大量槽位倾斜、热点槽位问题。



4、【建议】同一业务下的关联数据使用统一前缀，便于批量管理、数据清理与监控。

3.3 值规范

1、【强制】禁止存储超大Value（超过50KB必须审批）。

- 大 Value 会导致网络 IO、内存、命令执行效率急剧下降，引发卡顿、超时。
- 单条 Value 严格控制在 10KB 以内，hash、list、set、zset 元素个数不要超过 5000。

2、【强制】结构化数据（对象、数组、多字段数据）统一使用JSON格式序列化存储，禁止自定义特殊序列化格式，解析逻辑统一在应用层处理。

3、【强制】Value内容禁止存储明文敏感数据（手机号、身份证、密码、令牌等），敏感信息必须加密后再存入Redis。

4、【建议】避免Value冗余存储，非必要字段不要存入缓存，若为提升查询性能做冗余，仅限极少变更的静态字段。

- 字段筛选：只留必要字段，用独立 BO 接收缓存数据，拒绝 DAO 全量序列化。
- 动态字段：高频变更字段严禁冗余，单独查库或建独立缓存。
- 列表缓存：集合/列表仅存主键 ID，公共静态信息抽离为独立缓存。
- 更新逻辑：静态字段更新库后，只删缓存、不更缓存，靠过期/删除实现刷新。
- 评审卡点：代码评审强制检查字段冗余问题，线上定期巡检大 Value 与脏数据。

5、【建议】简单字符串、数字直接原生存储，无需额外序列化，减少编解码开销。

6、【建议】超长集合、列表类数据，进行拆分存储，禁止单Key承载数万条以上数据。

- 按页码/页号拆分（适合有序集合），一页对应一个独立 Key（{主前缀}:page:{页码}）。
- 按哈希桶拆分（适合无序集合/全量遍历），一个桶对应一个独立 Key（{主前缀}:bucket:{桶号}），提前预设固定桶数量（8/16/32 个桶），元素根据规则（根据元素 ID 取模分配桶）路由到不同桶。
- 按时间维度拆分（适合消息、日志、动态、操作记录等时序数据），按「天 / 小时」拆分 Key（{主前缀}:{时间戳}）。

3.4 数据类型选用规范

1、【强制】普通单值缓存（用户信息、配置、简单文本）：使用 String

2、【强制】对象/多字段结构化数据：

- 优先 String（JSON 格式）。
- 需单独更新某一字段时使用 Hash：Hash 适合少量字段、频繁局部更新场景，字段数量建议控制在 50 个以内。

3、【强制】有序列表、排行、时间线场景：使用ZSet（有序集合）。

4、【强制】无序集合、去重场景（标签、权限、用户组）：使用Set（集合）。

5、【强制】队列、栈、简单消息队列、顺序数据：使用List（列表）。



6、【强制】线上业务禁止使用Bitmap、HyperLogLog、Geo等特殊类型做核心业务存储，仅可用于统计、辅助类等非核心场景。

3.5 过期时间设计规范

- 1、【强制】所有业务缓存Key必须设置过期时间，禁止永久Key；过期时间根据业务热度合理设置。
- 2、【强制】过期时间单位统一使用秒，代码内禁止混用毫秒、分钟、小时单位。
- 3、【强制】热点缓存设置随机过期时间，避免大量Key同时过期引发缓存雪崩，比如：基础过期3600秒，随机增减0~300 秒。
- 4、【强制】临时数据、临时锁、临时令牌：过期时间设置为短时效（秒/分钟级），杜绝僵尸Key。
- 5、【建议】静态基础数据（字典、配置、地区码等）：过期时间建议4~12 小时，定时主动刷新。
- 6、【建议】动态数据（购物车、浏览记录）：过期时间建议1~4 小时，根据活跃度调整。
- 7、【强制】分布式锁必须设置过期时间，锁超时时间大于业务执行最大耗时，防止死锁。

4 缓存策略

- 1、【强制】通用缓存统一使用Cache-Aside（旁路缓存）模式。
 - 读：先查 Redis，未命中查数据库并回写缓存；
 - 写：先淘汰(delete)缓存，而不是更新(set)缓存，再删除/更新数据库。
- 2、【强制】针对高并发热点数据，主动做缓存预热，系统启动/数据变更后提前加载热点数据至Redis。
- 3、【建议】核心接口搭配布隆过滤器拦截非法不存在的业务ID。

5 淘汰策略

- 1、【强制】线上Redis内存淘汰策略配置为allkeys-lru，优先淘汰最近最少使用的Key。
- 2、【强制】监控Redis内存使用率，内存达到阈值前及时扩容、清理冷数据。

6 命令使用规范

6.1 强制禁用/限制命令

- 1、【强制】线上禁止使用高危命令：KEYS、FLUSHDB、FLUSHALL、CLIENT KILL。



- 批量查询/删除使用 SCAN 替代 KEYS。
- 2、【强制】禁止使用DEL批量删除海量Key，海量数据删除采用分批遍历 + 异步删除。
- 3、【强制】禁止在业务代码中使用事务、Lua脚本做复杂逻辑。
 - Lua 脚本仅限简单原子操作（分布式锁、简单计数），脚本长度精简，禁止长耗时脚本。
- 4、【强制】禁止使用阻塞命令（BRPOP/BLPOP等）做核心业务，易造成服务阻塞。

6.2 常规命令规范

- 1、【强制】批量操作优先使用批量命令（MGET/MSET），减少网络往返次数，禁止循环单次读写。
- 2、【强制】读写命令按需调用，禁止执行无用命令、频繁心跳式轮询查询。
- 3、【建议】集合、列表类操作，使用分页命令（LRANGE、SSCAN、ZSCAN），禁止一次性全量读取。
- 4、【建议】计数器场景优先使用原子自增命令INCR/INCRBY，避免应用层并发加锁。

6.3 客户端连接规范

- 1、【强制】统一团队Redis客户端框架，禁止项目内混用多种客户端。
- 2、【强制】配置合理的连接池参数：最大连接数、最小空闲连接、连接超时、空闲超时，避免连接耗尽、连接泄露。
- 3、【强制】代码层面必须手动释放连接，禁止连接不回收。
- 4、【强制】生产环境关闭Redis长连接，空闲连接自动回收。
- 5、【建议】添加重试机制，重试次数控制在3次以内，禁止无限重试。

6.4 运维与安全规范

- 1、【强制】必须设置密码，禁止空密码访问。
 - 密码定期轮换，禁止代码硬编码明文密码，统一配置中心管理。
- 2、【强制】Redis端口、访问IP做白名单限制，禁止公网直接暴露Redis服务。
- 3、【强制】开启日志监控，记录慢命令、大Value、异常连接、Key过期事件，慢命令阈值统一配置。
- 4、【强制】定期备份Redis数据，制定数据恢复预案，核心集群每日自动备份。
- 5、【强制】线上数据清理、实例重启、配置变更等运维操作必须提前报备，错开业务高峰，执行灰度操作。



6、【建议】对接监控平台，监控指标：内存使用率、CPU、连接数（全局监控连接数、空闲连接数）、命中率、过期Key数量、慢查询、集群状态。

7、【建议】缓存命中率维持在90%以上。

➤ 命中率过低及时排查 Key 设计、过期策略、业务逻辑问题。

7 工具

7.1 Redis Insight

可视化运维工具用于本地/测试环境数据查看、结构分析、性能调试，禁止用于生产直连。

■ 官网：[Redis Insight](https://redisinsight.io/)

8 附则

- 本规范自发布之日起实施。
- 本规范由鸟域道场进行解释。
- 本文件著作权归北京鸟域花香科研室所有，用户除认真学习外不得以任何理由在未经同意的情况下擅自将文件用于其他用途。

9 参考文献

《阿里 Redis 开发规范》