咕泡教育
—gupaoedu—

Redis如何提升程序处理性能

GUPAO EDU

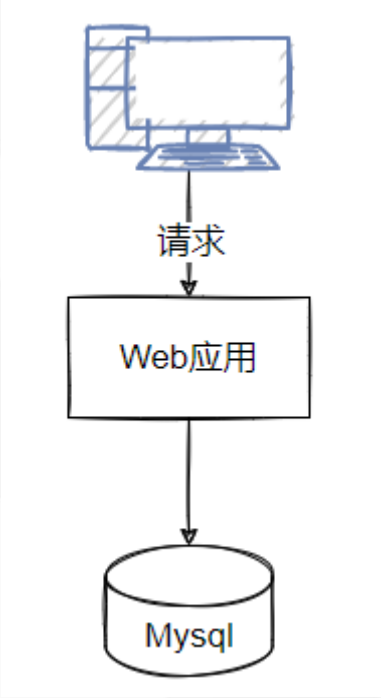
讲师：Mic

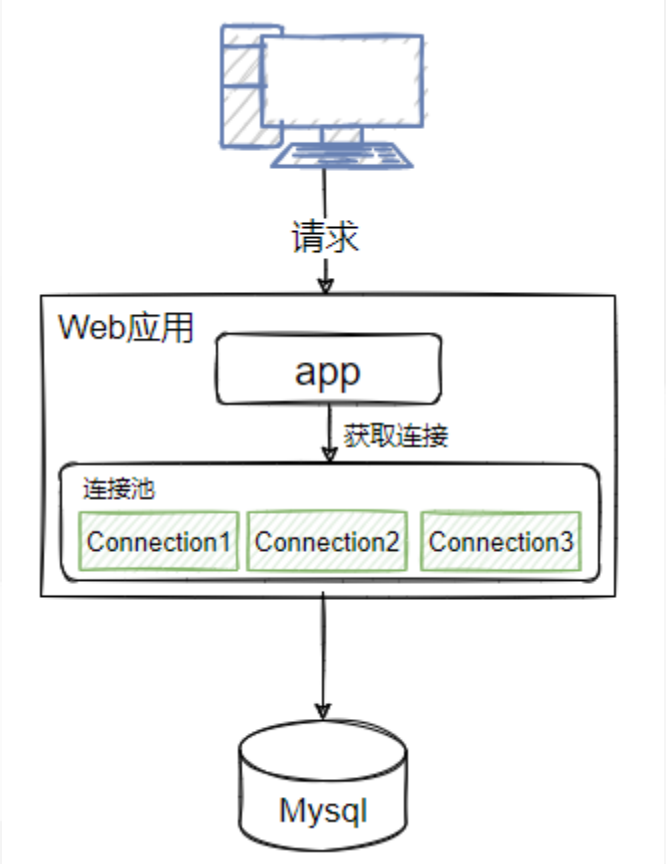
我们的愿景
推动每一次人才升级

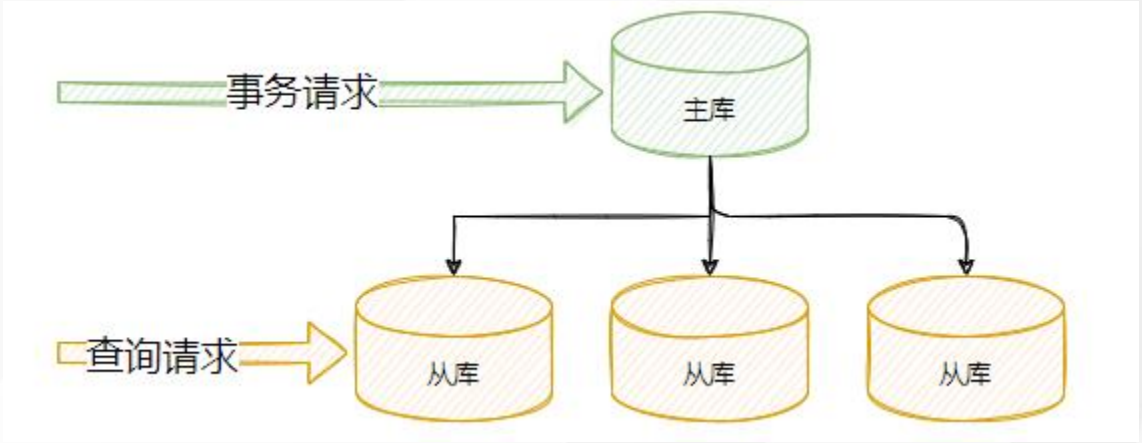
我们的使命
让每个人的职业生涯不留遗憾

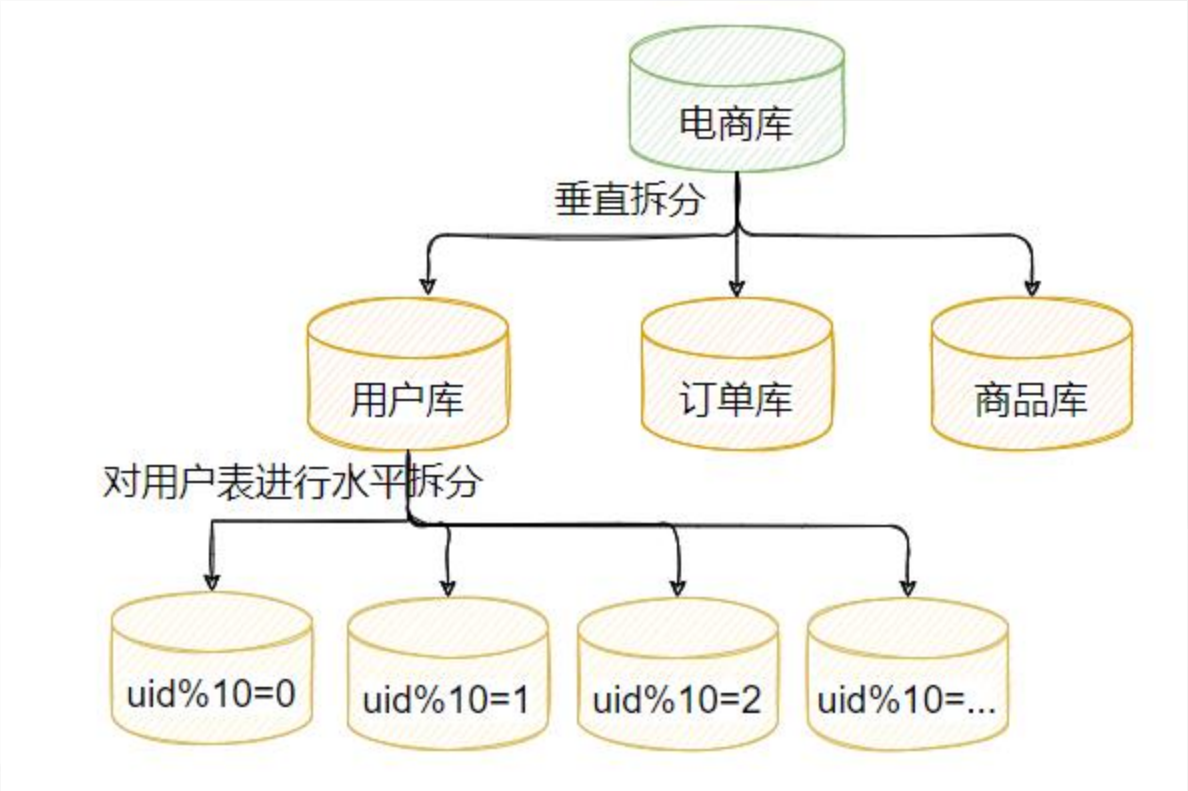
本节课程内容安排

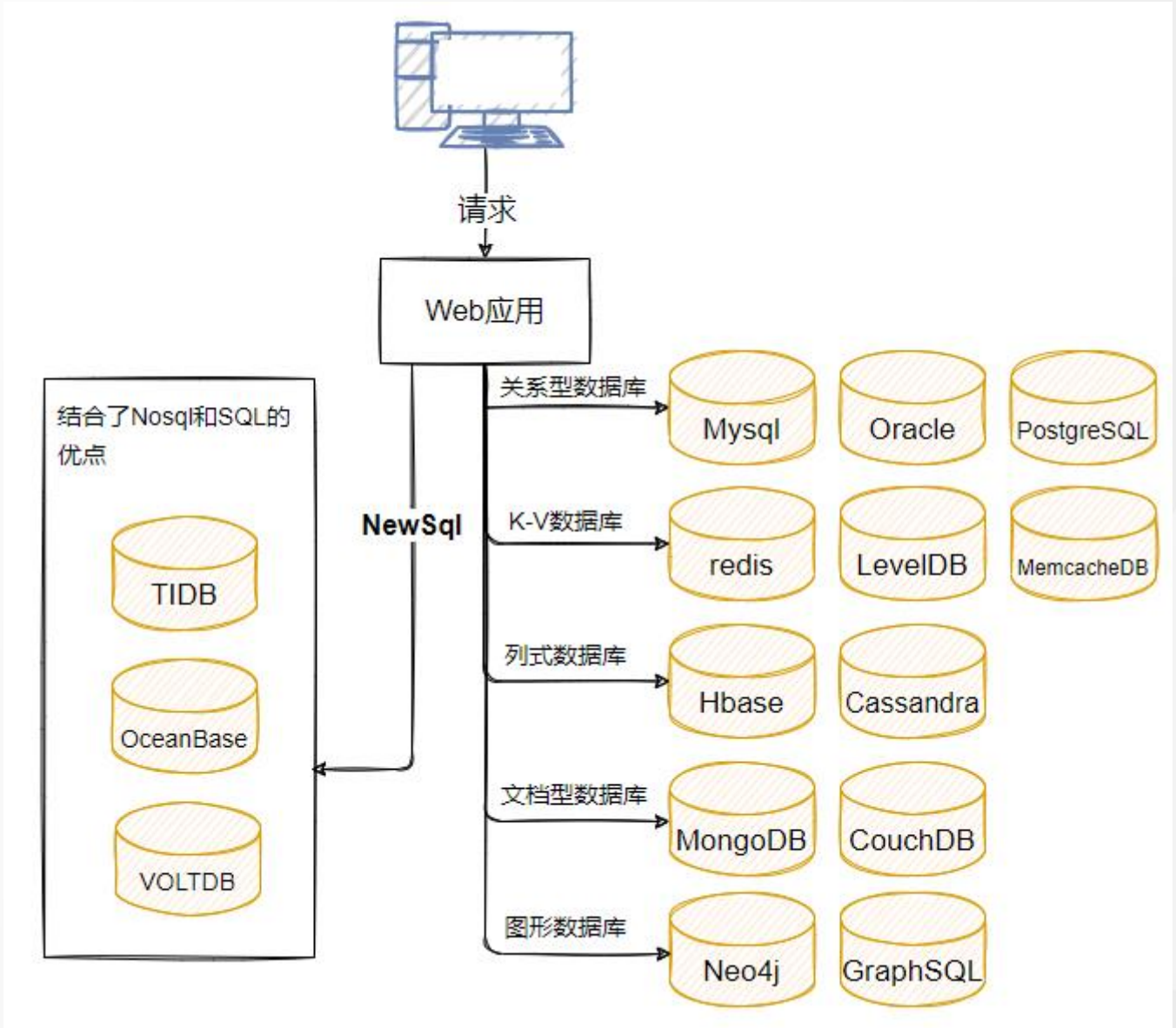
1. 数据库存储的性能问题
2. 不同存储技术带来的性能提升
3. K-V数据库的典型之Redis
4. 详解Redis的数据类型及场景


请求
Web应用
Mysql

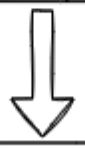

请求
Web应用
app
获取连接
连接池
Connection1
Connection2
Connection3
Mysql


事务请求
主库
查询请求
从库
从库
从库


电商库
垂直拆分
用户库
订单库
商品库
对用户表进行水平拆分
uid%10=0
uid%10=1
uid%10=2
uid%10=...

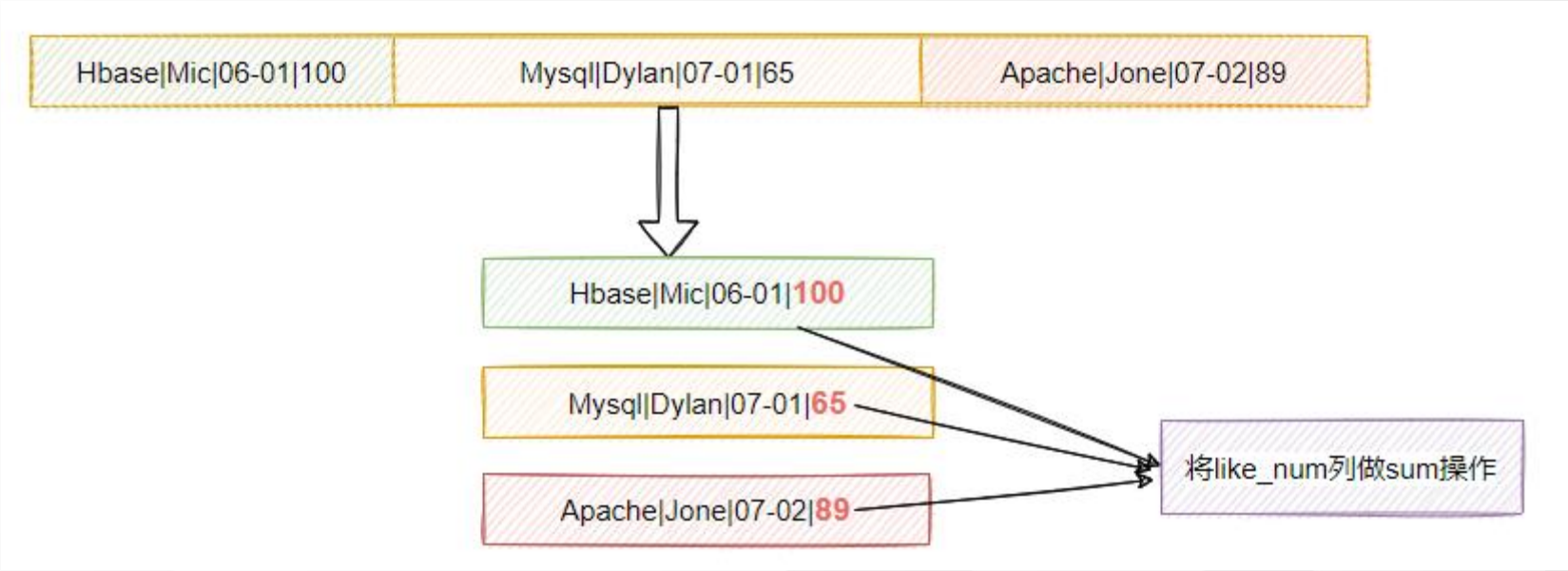

请求
Web应用
结合了Nosql和SQL的优点
TIDB
OceanBase
VOLTDB
NewSql
关系型数据库
Mysql
Oracle
PostgreSQL
K-V数据库
redis
LevelDB
MemcacheDB
列式数据库
Hbase
Cassandra
文档型数据库
MongoDB
CouchDB
图形数据库
Neo4j
GraphSQL

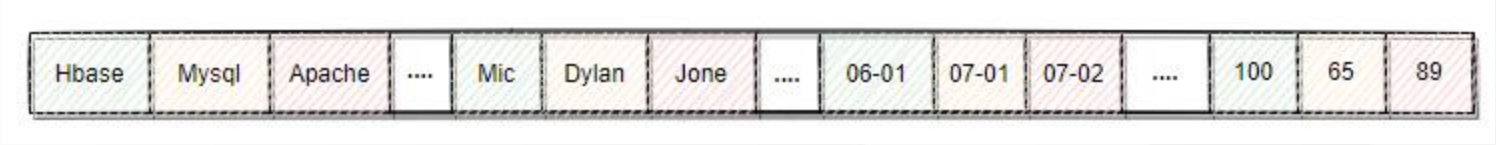
title	author	create_time	like_num
Hbase	Mic	06-01	100
Mysql	Dylan	07-01	65
Apache	Jone	07-02	89

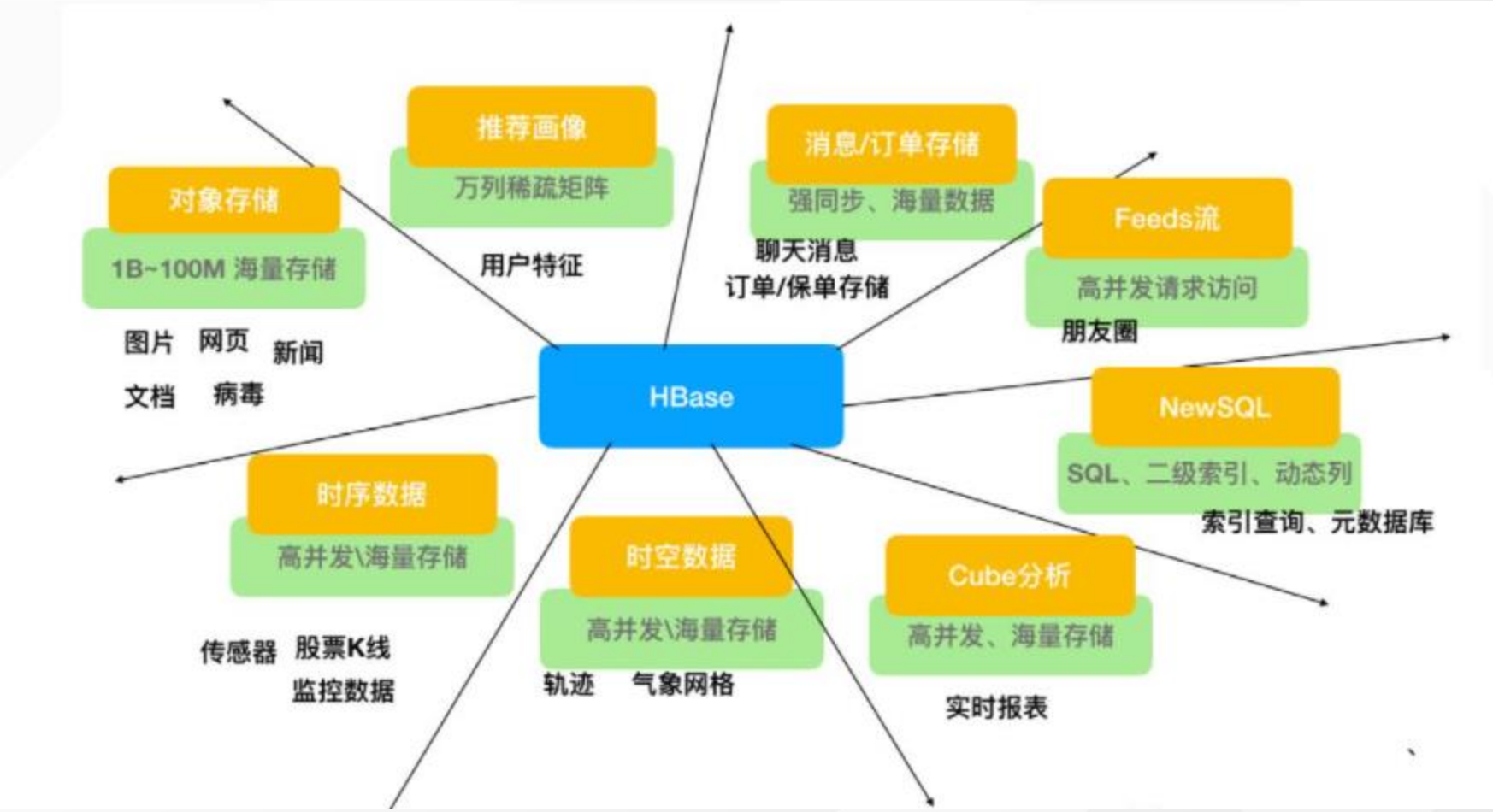


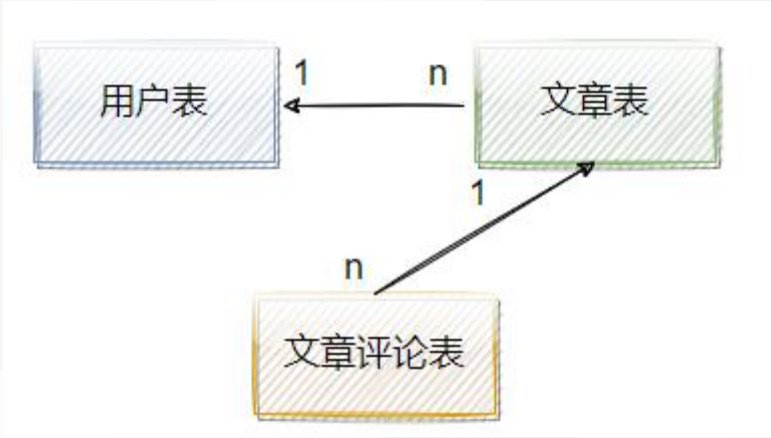
行存储的数据以下面这种方式存储在磁盘上

Hbase|Mic|06-01|100	Mysql|Dylan|07-01|65	Apache|Jone|07-02|89

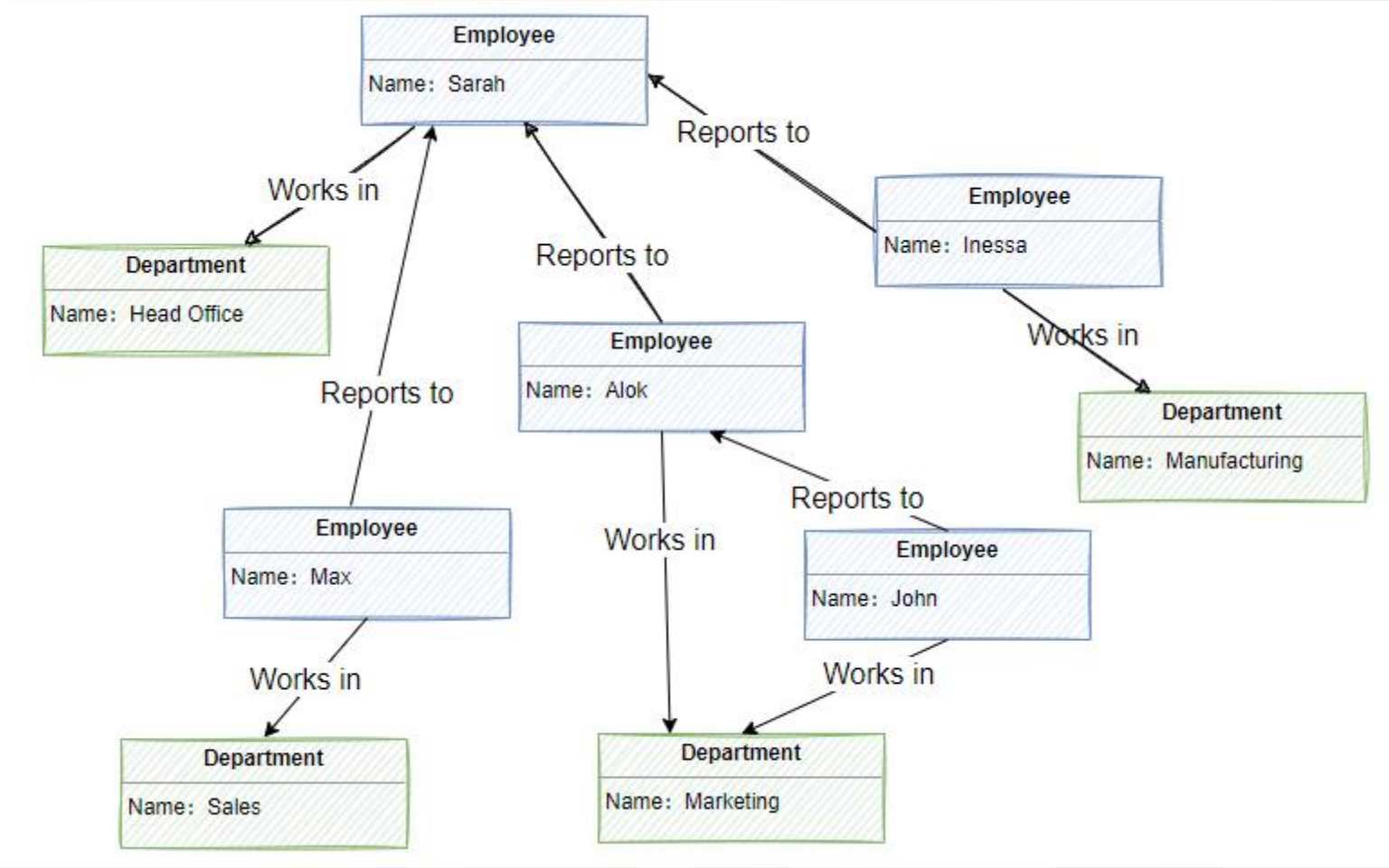

Hbase|Mic|06-01|100
Mysql|Dylan|07-01|65
Apache|Jone|07-02|89
Hbase|Mic|06-01|100
Mysql|Dylan|07-01|65
Apache|Jone|07-02|89
将like_num列做sum操作

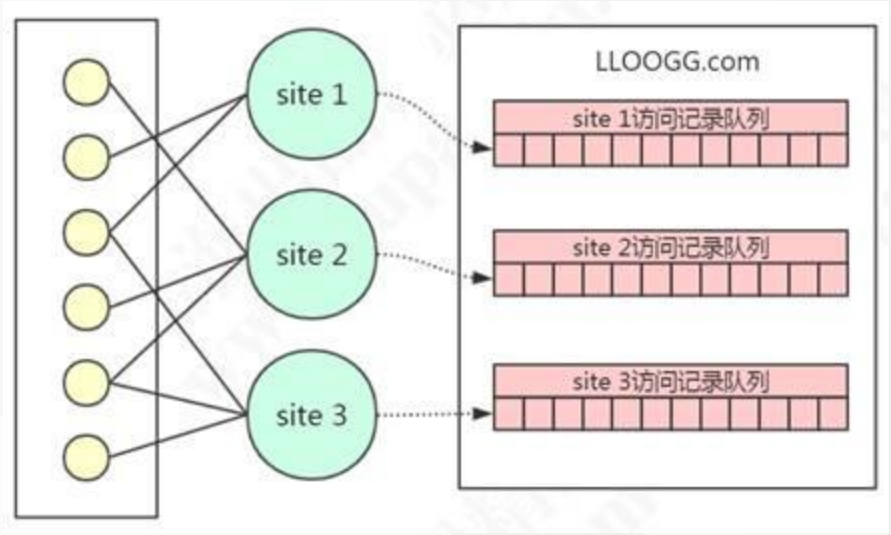

Hbase
Mysql
Apache
....
Mic
Dylan
Jone
....
06-01
07-01
07-02
....
100
65
89


HBase
对象存储
1B~100M 海量存储
图片 网页 新闻
文档 病毒
推荐画像
万列稀疏矩阵
用户特征
消息/订单存储
强同步、海量数据
聊天消息
订单/保单存储
Feeds流
高并发请求访问
朋友圈
NewSQL
SQL、二级索引、动态列
索引查询、元数据库
时序数据
高并发\海量存储
传感器 股票K线
监控数据
时空数据
高并发\海量存储
轨迹 气象网格
Cube分析
高并发、海量存储
实时报表


用户表
1
n
文章表
1
n
文章评论表


{
_id: "5cf0029caff5056591b0ce7d",
firstname: 'Jane',
lastname: 'Wu',
address: {
street: '1 Circle Rd',
city: 'Los Angeles',
state: 'CA',
zip: '90404'
}
}


Employee
Name: Sarah
Reports to
Works in
Employee
Name: Inessa
Department
Name: Head Office
Reports to
Works in
Employee
Name: Alok
Department
Name: Manufacturing
Reports to
Reports to
Works in
Employee
Name: Max
Employee
Name: John
Works in
Works in
Department
Name: Sales
Department
Name: Marketing


site 1
site 2
site 3
LLOOGG.com
site 1访问记录队列
site 2访问记录队列
site 3访问记录队列

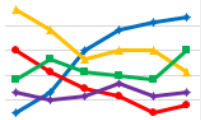
K-V数据库使用排名

DB-Engines Ranking of Key-value Stores

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

This is a partial list of the complete ranking showing only key-value stores.

Read more about the method of calculating the scores.



trend chart

☐ include secondary database models

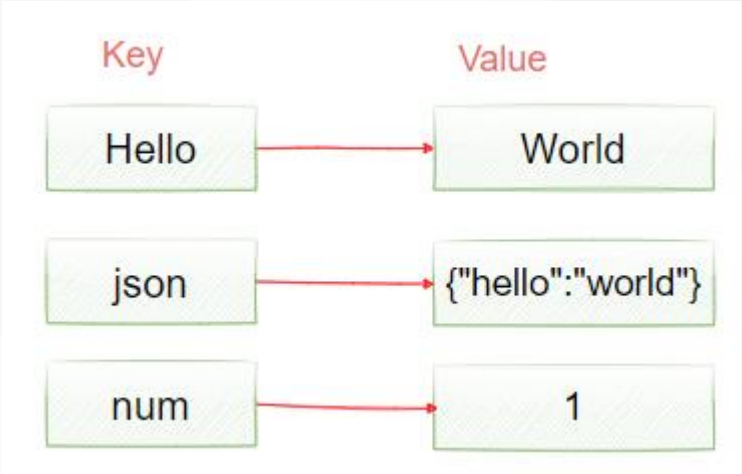
63 systems in ranking, June 2021

Rank Jun 2021	Rank May 2021	Rank Jun 2020	DBMS	Database Model	Score Jun 2021	Score May 2021	Score Jun 2020
1.	1.	1.	Redis	Key-value, Multi-model	165.25	+3.08	+19.61
2.	2.	2.	Amazon DynamoDB	Multi-model	73.76	+3.69	+8.90
3.	3.	3.	Microsoft Azure Cosmos DB	Multi-model	36.47	+1.76	+5.67
4.	4.	4.	Memcached	Key-value	25.18	+0.68	+0.37
5.	5.	↑ 6.	etcd	Key-value	10.22	+0.80	+2.17
6.	6.	↓ 5.	Hazelcast	Key-value, Multi-model	9.37	+0.19	+0.96
7.	7.	↑ 8.	Ehcache	Key-value	7.49	+0.26	+1.21
8.	8.	↓ 7.	Aerospike	Key-value, Multi-model	5.77	+0.86	-0.89
9.	9.	↑ 10.	Riak KV	Key-value	5.40	+0.82	+0.40
10.	10.	↑ 11.	Ignite	Multi-model	4.93	+0.54	+0.06
11.	11.	↓ 9.	ArangoDB	Multi-model	4.92	+0.53	-0.47
12.	12.	12.	OrientDB	Multi-model	4.45	+0.26	-0.37
13.	13.	13.	Oracle NoSQL	Multi-model	4.31	+0.61	+0.09
14.	14.	↑ 17.	RocksDB	Key-value	3.58	+0.49	+0.72
15.	15.	↓ 14.	InterSystems Caché	Multi-model	3.24	+0.34	-0.22
16.	↑ 17.	↓ 15.	Oracle Berkeley DB	Multi-model	3.08	+0.44	-0.12
17.	↓ 16.	↑ 18.	Infinispan	Key-value	2.95	+0.20	+0.08
18.	↑ 19.	↓ 16.	LevelDB	Key-value	2.80	+0.44	-0.26
19.	↑ 21.	19.	ScyllaDB	Multi-model	2.71	+0.57	0.00
20.	↓ 18.	20.	Oracle Coherence	Key-value	2.50	+0.06	+0.03

Redis的数据类型


String(字符串)
Hash(哈希表)
HyperLogLog(基数)
List(有序且可重复集合)
Set(无序且补课重复集合)
Streams(流信息)
zSet(有序且补课重复集合)
Geospatial(地理位置计算)
Bit Arrays(位集合)

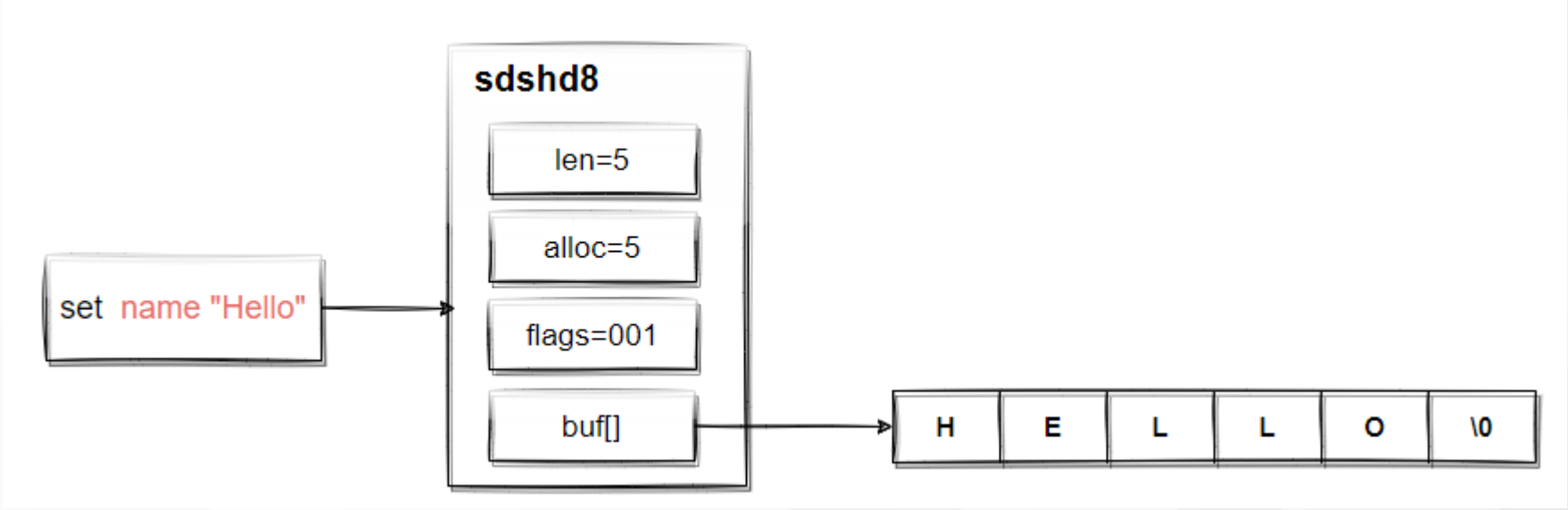
String类型



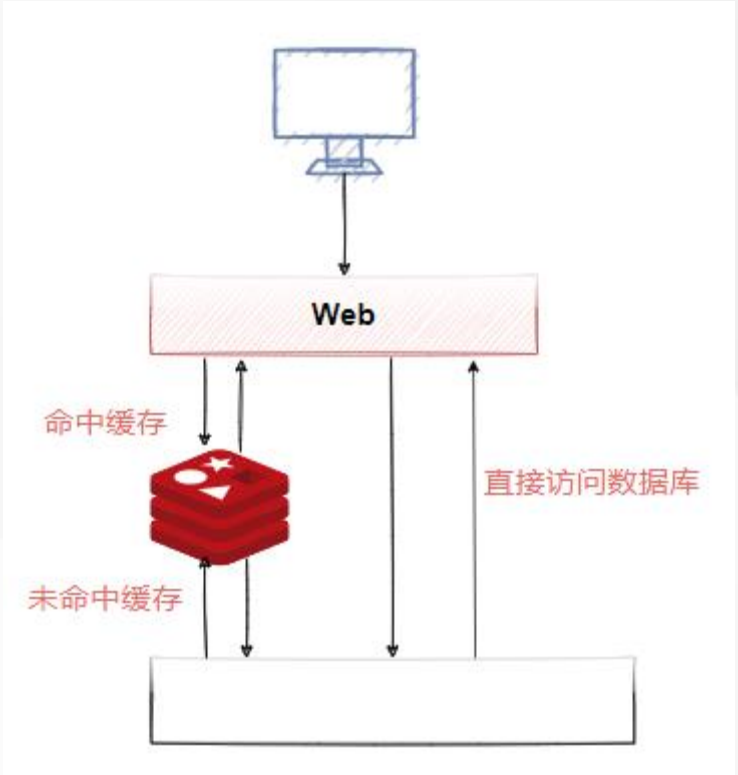
String类型常用命令

命令	描述
set key value [ex seconds] ...	设置指定key的value，如果key不存在则创建，否则，修改
get key	获取指定key的value值
del key [key ...]	删除指定key
mset key value [key value ...]	批量设置值
mget key [key ...]	批量获取值
incr key	对指定的key对应的value值进行原子递增（value必须是int类型）
decr key	对指定的key对应的value值进行原子递减
setex key value(second)	设置指定key的过期时间，单位为秒
setnx	将key的值设置为value，如果key存在，返回0不做任何处理，否则返回1
getset	将指定key的值设置为value，并返回key修改之前的值

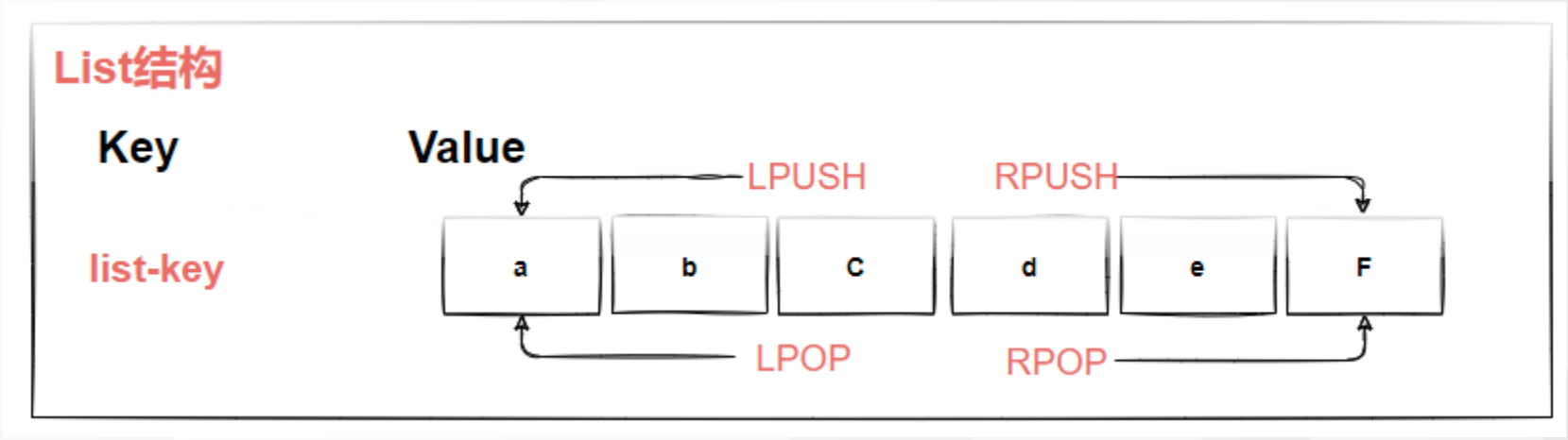
SDS类型存储



String类型应用场景



List存储结构



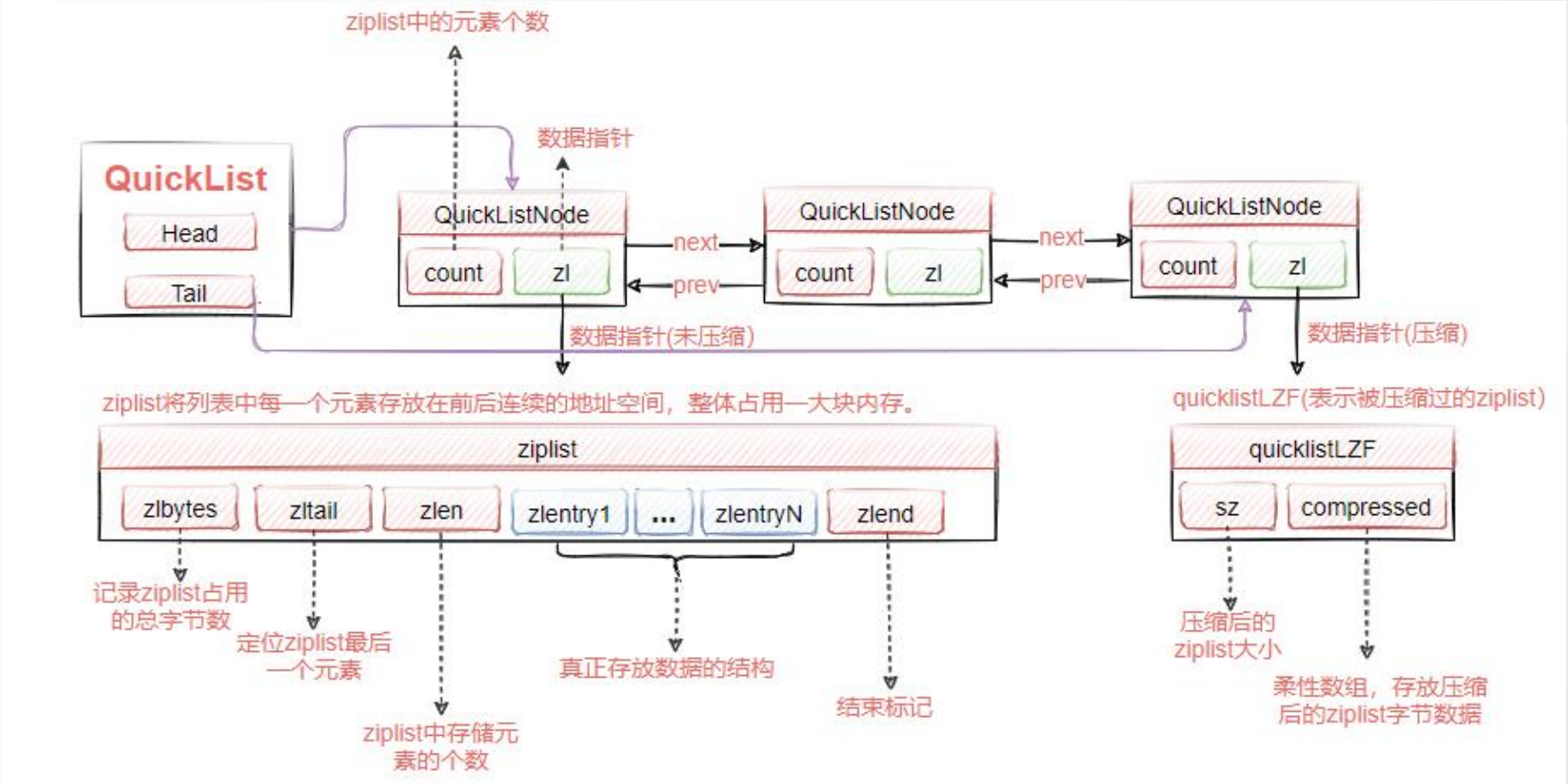
咕泡教育
—gupaoedu—

List常用指令

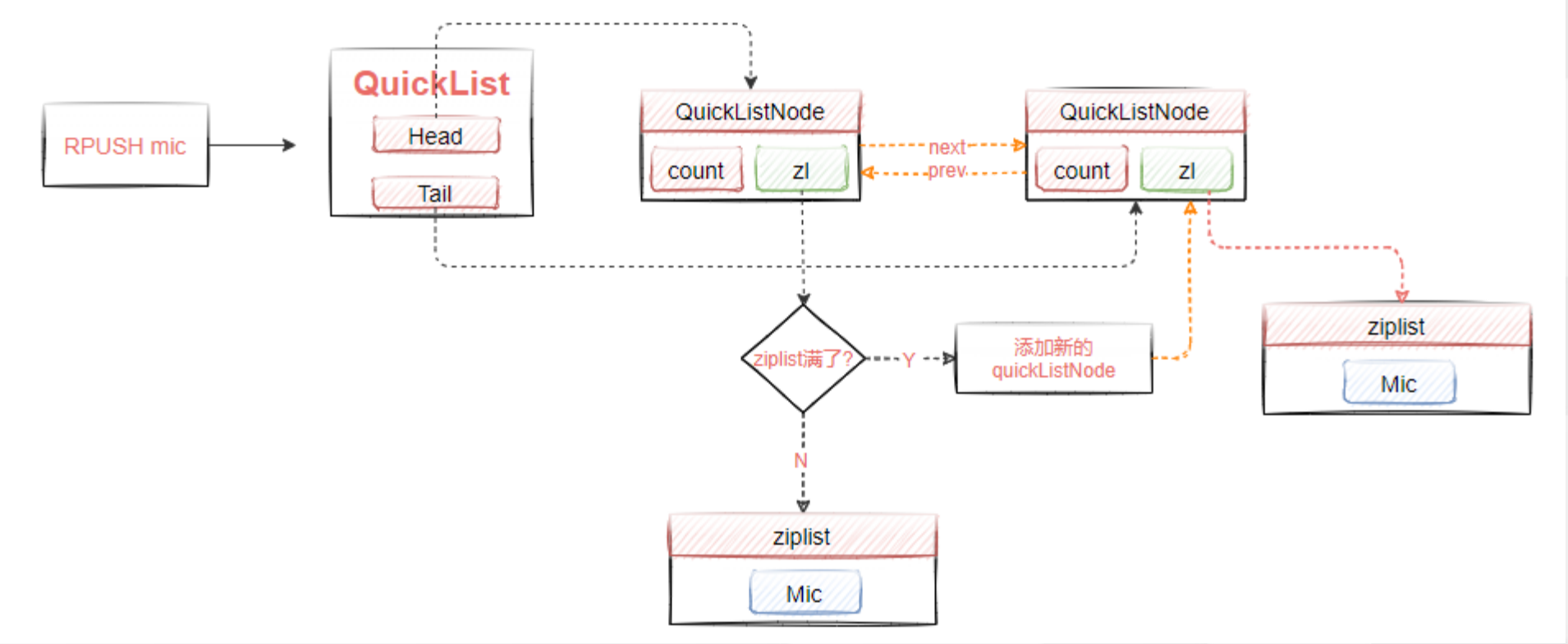
命令	描述
LPUSH key value [value...]	从队列的左边入队一个或多个元素
LPOP key	从队列的左边出队一个元素
RPUSH key value [value ...]	从队列的右边入队一个元素
RPOP key	从队列的右边出队一个元素
BLPOP key [key ...] timeout	删除，并获得该列表中的第一元素，如果当前队列没有元素则阻塞，直到有新的元素
BRPOP key [key ...] timeout	删除，并获得该列表中的最后一个元素，如果当前队列没有元素则阻塞，直到有新的元素
LRANGE key start stop	返回列表 key 中指定区间内的元素，区间以偏移量 start 和 stop 指定。
RPOPLPUSH source destination	命令 RPOPLPUSH 在一个原子时间内，执行以下两个动作： 1. 将列表 source 中的最后一个元素(尾元素)弹出，并返回给客户端。 2. 将 source 弹出的元素插入到列表 destination ，作为 destination 列表的的头元素。
LLEN key	返回列表 key 的长度。

List数据结构-压缩列表

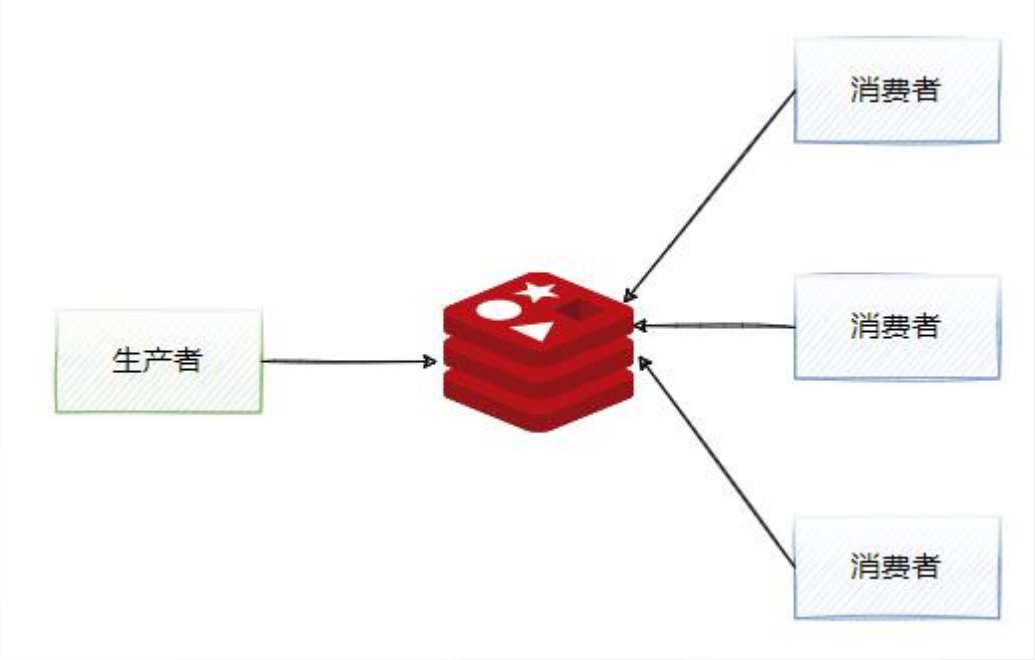
List数据结构


ziplist中的元素个数
数据指针
QuickList
Head
Tail
QuickListNode
count
zl
next
prev
QuickListNode
count
zl
next
prev
QuickListNode
count
zl
数据指针(未压缩)
数据指针(压缩)
ziplist将列表中每一个元素存放在前后连续的地址空间，整体占用一大块内存。
quicklistLZF(表示被压缩过的ziplist)
ziplist
zlbytes
zltail
zlen
zlentry1
...
zlentryN
zlend
quicklistLZF
sz
compressed
记录ziplist占用的总字节数
定位ziplist最后一个元素
ziplist中存储元素的个数
真正存放数据的结构
结束标记
压缩后的ziplist大小
柔性数组，存放压缩后的ziplist字节数据

List数据结构原理


RPUSH mic
QuickList
Head
Tail
QuickListNode
count
zl
next
prev
QuickListNode
count
zl
ziplist满了?
Y
添加新的
quickListNode
ziplist
Mic
N
ziplist
Mic

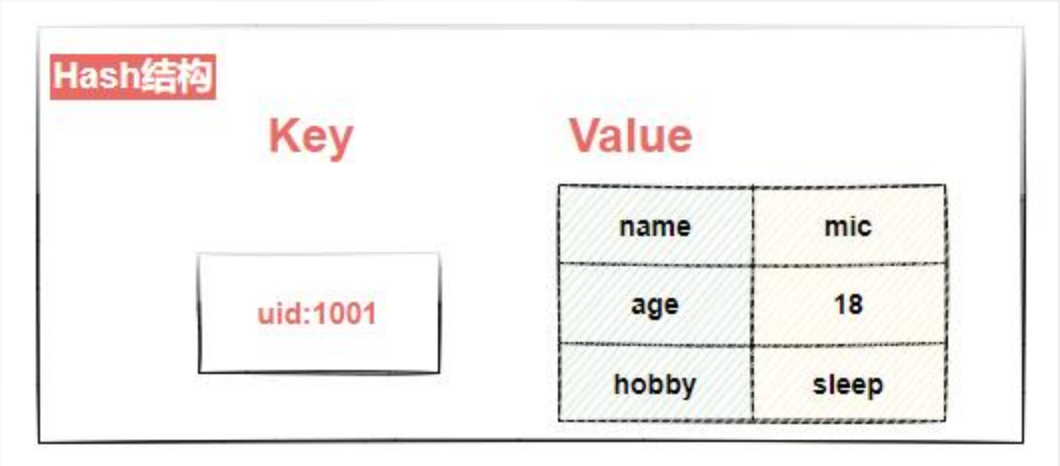
List应用场景-消息队列



List应用场景-红包


发红包
创建红包
10元/10个
在发送前，先进行计算，得到10个红包数据
保存到list中(RPUSH)
List
1元
0.8元
0.2元
4元
0.5元
1.6元
0.9元
0.4元
0.4元
0.2元
抢红包
抢红包（LPOP）

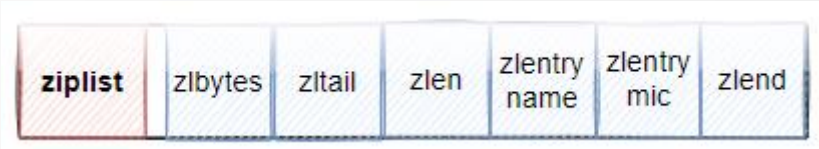
Hash结构

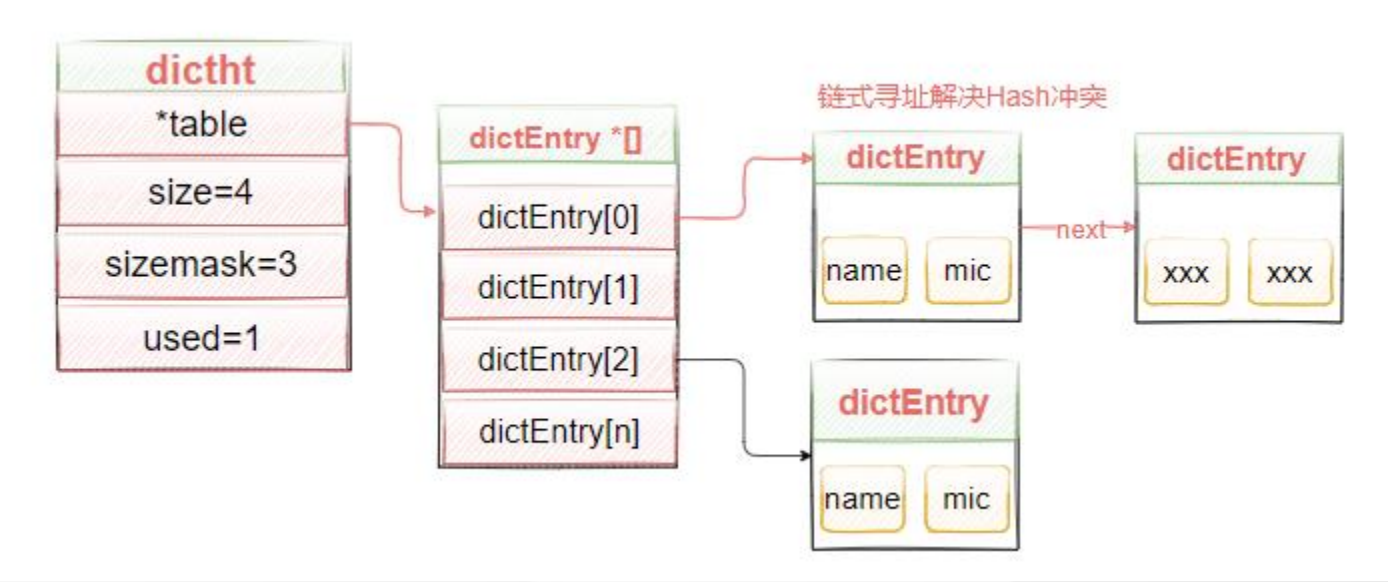


Hash结构常用命令

命令	描述	用法
HSET	将Hash表key中field字段的值设置为value 如果key不存在，则创建一个新的hash表，否则覆盖field的值	HSET key field value
HGET	返回Hash表key中指定field字段的值	HGET key field
HDEL	删除Hash表Key中一个或者多个field，如果不存在，则直接忽略	HDEL key field [field ...]
HEXISTS	查看Hash表key中，指定field是否存在，存在返回1，否则返回0	HEXISTS key field
HGETALL	返回Hash表key中所有的field和value	HGETALL key
HKEYS	返回Hash表key中所有的field	HKEYS key
HLEN	返回Hash表key中field的数量	HLEN key
HVALS	返回Hash表key中所有的field的value	HVALS key
HMGET	返回Hash表Key中，一个或多个指定field的值	HMGET key field [field ...]
HMSET	同时将多个field-value设置到Hash表指定的key中，如果存在field，则覆盖。如果key不存在，则创建一个新的Hash表	HMSET key field value [field value ...]

Hash存储结构

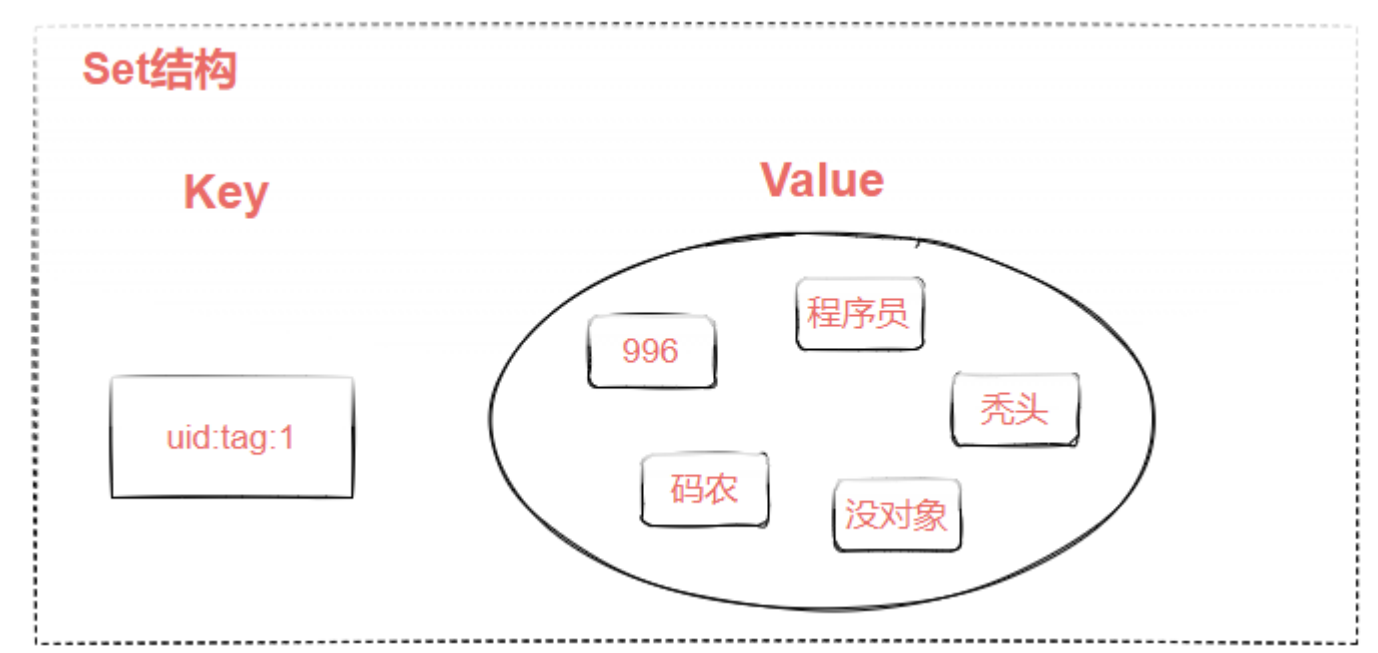

ziplist
zlbytes
zltail
zlen
zlentry
name
zlentry
mic
zlend


dictht
*table
size=4
sizemask=3
used=1
dictEntry *[]
dictEntry[0]
dictEntry[1]
dictEntry[2]
dictEntry[n]
链式寻址解决Hash冲突
dictEntry
name
mic
next
dictEntry
xxx
xxx
dictEntry
name
mic

Hash实际应用场景

1. 缓存对象数据
2. 缓存购物车信息
3. 缓存商品详情
4. 做计数器

Set类型

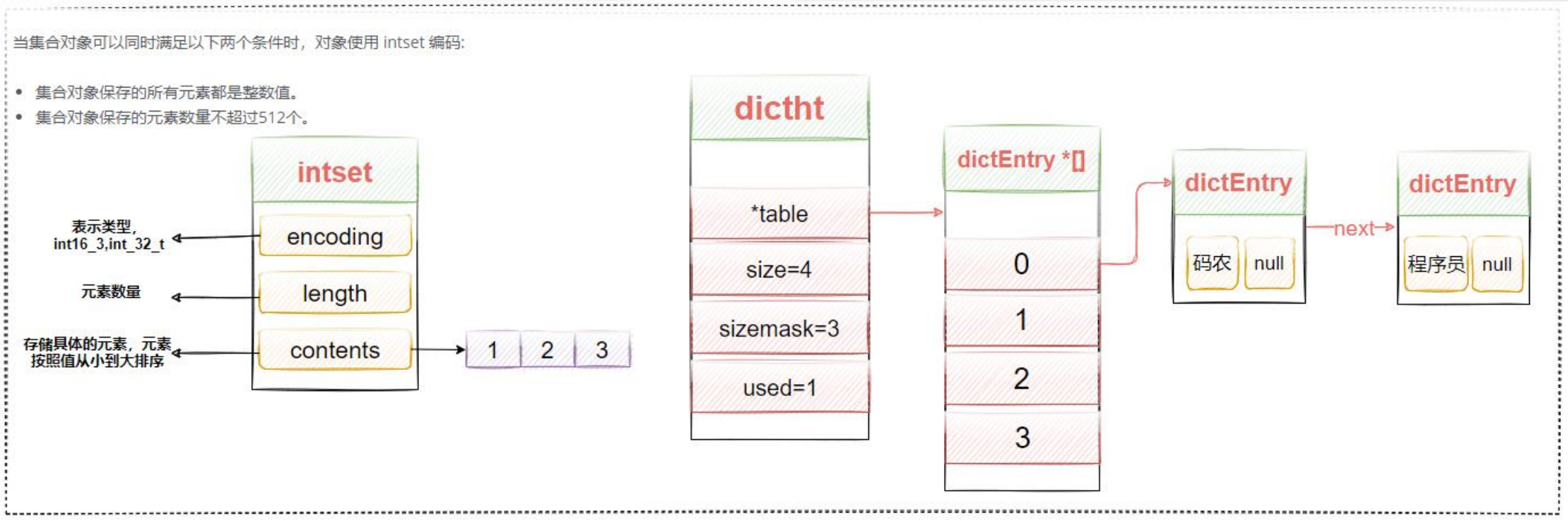


Set类型操作指令

命令	说明	时间复杂度
SADD key member [member ...]	添加一个或者多个元素到集合(set)里	O(N)
SCARD key	获取集合里面的元素数量	O(1)
SDIFF key [key ...]	获得队列不存在的元素	O(N)
SDIFFSTORE destination key [key ...]]	获得队列不存在的元素，并存储在一个关键的结果集	O(N)
SINTER key [key ...]	获得两个集合的交集	O(N*M)
SINTERSTORE destination key [key ...]	获得两个集合的交集，并存储在一个关键的结果集	O(N*M)
SISMEMBER key member	确定一个给定的值是一个集合的成员	O(1)
SMEMBERS key	获取集合里面的所有元素	O(N)
SMOVE source destination member	移动集合里面的一个元素到另一个集合	O(1)
SPOP key [count]	删除并获取一个集合里面的元素	O(1)
SRANDMEMBER key [count]	从集合里面随机获取一个元素	
SREM key member [member ...]]	从集合里删除一个或多个元素	O(N)
SUNION key [key ...]]	添加多个set元素	O(N)
SUNIONSTORE destination key [key ...]	合并set元素，并将结果存入新的set里面	O(N)

咕泡教育
—gupaoedu—

Set类型存储结构

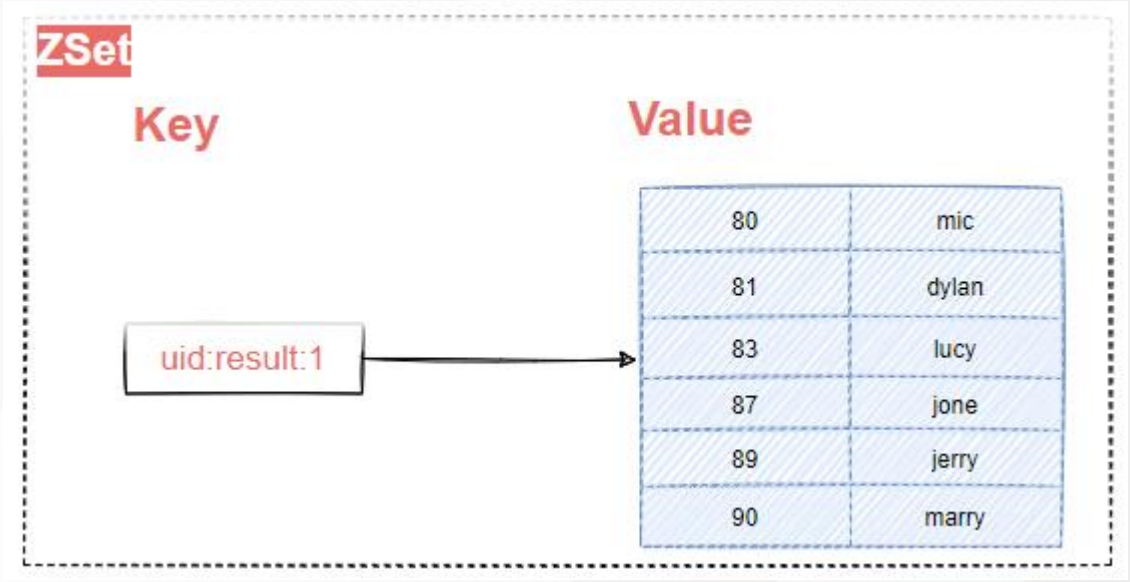


咕泡教育
—gupaoedu—

Set类型使用场景

1. 用户标签设置
2. 相关商品信息展示

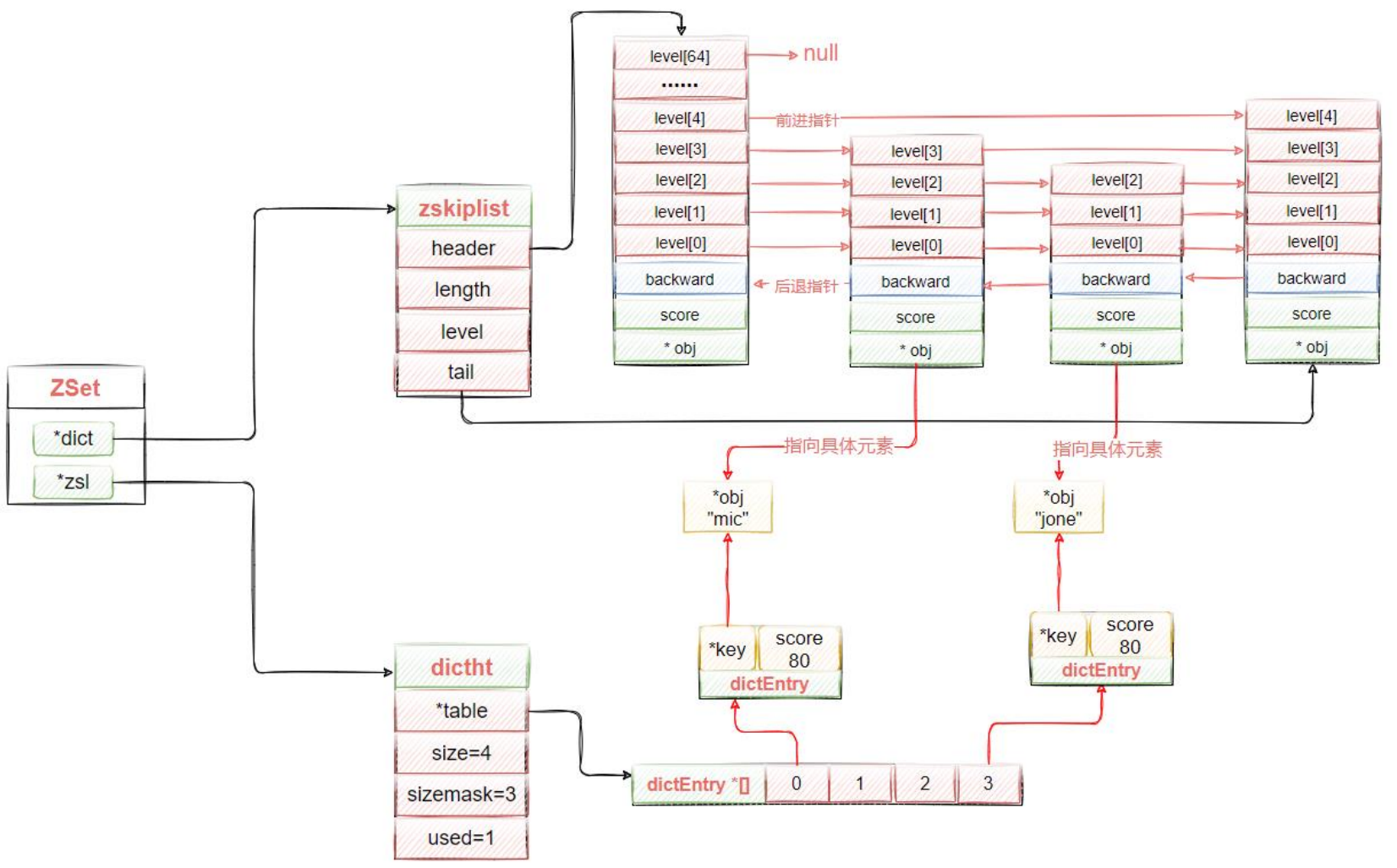
Zset类型


ZSet
Key
Value
uid:result:1
80
mic
81
dylan
83
lucy
87
jone
89
jerry
90
marry

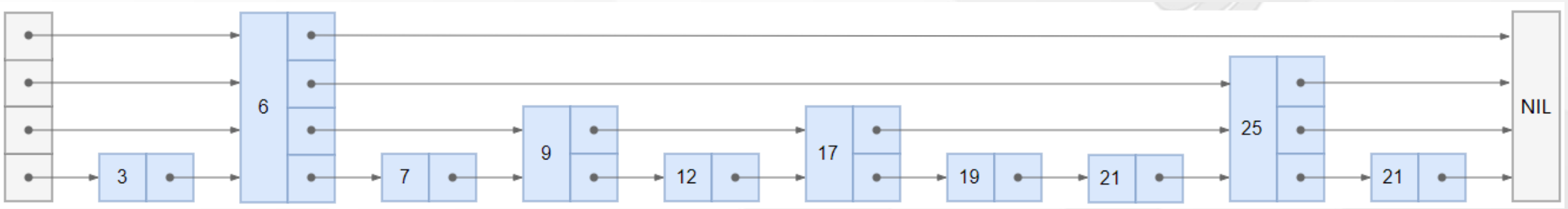
Zset类型操作指令

命令	描述
zadd key score member[{score member}...]	创建或设置指定key对应的有序集合，根据每个值对应的score来排名，升序。 例如有命令 zadd key1 10 A 20 B 30 D 40 C; 那么真实排名是 A B D C
zrem key member	删除指定key对应的集合中的member元素
zcard key	返回指定key对应的有序集合的元素数量
zincrby key increment member	将指定key对应的集合中的member元素对应的分值递增加 increment
zcount key min max	返回指定key对应的有序集合中，分值在min~max之间的元素个数
zrank key member	返回指定key对应的有序集合中，指定元素member在集合中排名，从0开始切分值是从小到大升序
zscore key member	返回指定key中的集合中指定member元素对应的分值
zrange key min max [withscores]	返回指定key对应的有序集合中，索引在min~max之间的元素信息，如果带上 withscores 属性的话，可以将分值也带出来
zrevrank key member	返回指定key对应的集合中，指定member在其中的排名，注意排名从0开始且按照分值从大到小降序
zrevrange key start end [withscores]	指定key对应的集合中，分值在 start~end之间的降序，加上 withscores 的话可以将分值以及value都显示出来
zrangebyscore key start end [withscores]	同 zrange命令不同的是，zrange命令是索引在startend范围的查询，而zrangebyscore命令是根据分值在startend之间的查询且升序展示
zremrangebyrank key start end	移除指定key对应集合中索引在start~end之间(包括start和end本身)的元素
zremrangebyscore by min max	同zremrangebyrank命令类似，不同的该命令是删除分值在min~max之间的元素

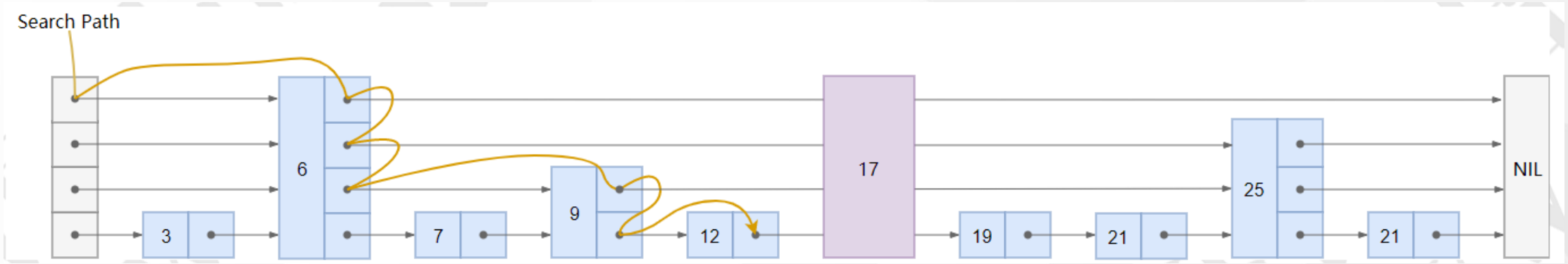
Zset类型存储结构


level[64]
null
.....
level[4]
level[3]
level[2]
level[1]
level[0]
backward
score
* obj
前进指针
后退指针
zskiplist
header
length
level
tail
ZSet
*dict
*zsl
dictht
*table
size=4
sizemask=3
used=1
dictEntry *[]
0
1
2
3
指向具体元素
指向具体元素
*obj
"mic"
*obj
"jone"
*key
score
80
dictEntry

关于跳跃表



关于跳跃表



Zset使用场景

1. 排行榜
2. 热点话题

咕泡教育
—gupaoedu—

谢谢观赏

GUPAO EDU

Mic

我们的愿景

推动每一次人才升级

我们的使命

让每个人的职业生涯不留遗憾