

## Commands

### Introduction

### Strings

### Lists

### Sets

### Sorted Sets (ZSets)

#### SORTED SETS Introduction

- ZADD
- ZINCRBY
- ZRANGE
- ZRANGESTORE 6.2
- ZLS Ent
- ZREVRANGE
- ZRANGEBYSCORE
- ZREVRANGEBYSCORE
- ZRANGEBYLEX
- ZREVRANGEBYLEX
- ZCARD
- ZLEN Ent
- ZCOUNT
- ZLEXCOUNT
- ZRANK
- ZREVRANK
- ZSCORE
- ZMSCORE 6.2
- ZREM
- ZRM Ent
- ZREMRANGEBYRANK
- ZREMRANGEBYSCORE
- ZREMRANGEBYLEX
- ZUNION 6.2
- ZINTER 6.2
- ZDIFF 6.2
- ZUNIONSTORE
- ZINTERSTORE
- ZDIFFSTORE 6.2
- ZSCAN
- ZPOPMIN 5.0
- ZPOPMAX 5.0
- BZPOPMIN 5.0
- BZPOPMAX 5.0
- ZSMEMBER Ent
- ZRANDMEMBER 6.2

### Hashes

### Streams

### Common Keys

### Recyclebin New

### Bits

### Pub/Sub

### Lua Script

### HyperLogLog(PF)

### Geo

### Connection

### Redis on Windows

# SORTED SETS Introduction



레디스 개발자 교육 신청



레디스 정기점검/기술지원  
Redis Technical Support



레디스 엔터프라이즈 서버  
Redis Enterprise Server

## 소개

Sorted Sets는 key 하나에 여러개의 score와 value로 구성됩니다.

Value는 score로 sort되며 중복되지 않습니다.

score가 같으면 value로 sort됩니다.

Sorted Sets에서는 집합이라는 의미에서 value를 member라 부릅니다.

Sorted Sets은 주로 sort가 필요한 곳에 사용됩니다.

## 명령어 요약

- **SET:** ZADD
- **GET:** ZRANGE, ZRANGEBYSCORE, ZRANGEBYLEX, ZREVRANGE, ZREVRANGEBYSCORE, ZREVRANGEBYLEX, ZRANK, ZREVRANK, ZSCORE, ZCARD, ZCOUNT, ZLEXCOUNT, ZSCAN
- **POP:** ZPOPMIN, ZPOPMAX
- **REM:** ZREM, ZREMRANGEBYRANK, ZREMRANGEBYSCORE, ZREMRANGEBYLEX
- **INCR:** ZINCRBY
- **집합연산:** ZUNIONSTORE, ZINTERSTORE
- **Enterprise:** ZSMEMBER, ZLS, ZRM, SLEN

## Sorted Sets 명령어 리스트

Commands	Version	Syntax	Description
ZADD	1.2.0	key score member [score member ...]	집합에 score와 member를 추가
ZCARD	1.2.0	key	집합에 속한 member의 갯수를 조회
ZINCRBY	1.2.0	key increment member	지정한 만큼 score 증가, 감소
ZRANGE	1.2.0	key start stop [withscores]	index로 범위를 지정해서 조회
ZRANGEBYSCORE	1.2.0	key min max [withscores] [limit offset count]	score로 범위를 지정해서 조회
ZREM	1.2.0	key member [member ...]	집합에서 member를 삭제
ZREMRANGEBYSCORE	1.2.0	key min max	score로 범위를 지정해서 member를 삭제
ZREVRANGE	1.2.0	key start stop [withscores]	index로 범위를 지정해서 큰 것부터 조회

ZSCORE	1.2.0	key member	member를 지정해서 score를 조회
ZCOUNT	2.0.0	key min max	score로 범위를 지정해서 갯수 조회
ZRANK	2.0.0	key member	member를 지정해서 rank(index)를 조회
ZREVRANK	2.0.0	key member	member를 지정해서 reverse rank(index)를 조회
ZREMRANGEBYRANK	2.0.0	key start stop	index로 범위를 지정해서 member를 삭제
ZUNIONSTORE	2.0.0	dest_key numkeys src_key [src_key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM   MIN   MAX]	합집합을 구해서 새로운 집합에 저장
ZINTERSTORE	2.0.0	dest_key numkeys src_key [src_key ...] [WEIGHTS weight [weight ...]] [AGGREGATE SUM   MIN   MAX]	교집합을 구해서 새로운 집합에 저장
ZREVRANGEBYScore	2.2.0	key max min [withscores] [limit offset count]	score로 범위를 지정해서 큰 것부터 조회
ZSCAN	2.8.0	key cursor [MATCH pattern] [COUNT count]	score, member를 일정 단위 갯수 만큼씩 조회
ZRANGEBYLEX	2.8.9	key min max [limit offset count]	member로 범위를 지정해서 조회
ZLEXCOUNT	2.8.9	key min max	member로 범위를 지정해서 갯수 조회
ZREMRANGEBYLEX	2.8.9	key min max	member로 범위를 지정해서 member를 삭제
ZREVRANGEBYLEX	2.8.9	key max min [limit offset count]	member로 범위를 지정해서 큰 것부터 조회
ZPOPMIN	5.0.0	key	작은 값부터 꺼내온다
ZPOPMAX	5.0.0	key	큰 값부터 꺼내온다
BZPOPMIN	5.0.0	key	데이터가 들어오면 작은 값부터 꺼내온다
BZPOPMAX	5.0.0	key	데이터가 들어오면 큰 값부터 꺼내온다
ZMSCORE	6.2.0	member [member ...]	member의 score를 리턴 - 여러 개 가능
ZRANDMEMBER	6.2.0	key	임의(random)의 멤버를 조회
ZRANGESTORE	6.2.0	dst src start stop	조회해서 다른 키에 저장
ZUNION	6.2.0	numkeys key [key ...]	합집합을 구함
ZINTER	6.2.0	numkeys key [key ...]	교집합을 구함
ZDIFF	6.2.0	numkeys key [key ...]	차집합을 구함
ZDIFFSTORE	6.2.0	destination numkeys key [key ...]	차집합을 구해서 새로운 집합에 저장
ZISMEMBER	Ent 6.0.0	key member	집합에 member가 존재하는지 확인
ZLS	Ent 6.0.0	key pattern	패턴(pattern)으로 값(value) 조회
ZRM	Ent 6.0.0	key pattern	패턴(pattern)으로 값(value) 삭제

Total : 36

---

## Redis Internal : Sorted Set 내부 데이터 구조

### Sorted Set 데이터 구조

- **스킵 리스트(SKIP LIST):** Sorted Set의 메인 데이터 구조인 스킵 리스트를 알면,
  - 이제 우리는 눈을 감고도 ZADD가 어떻게 동작하는지,
  - ZRANGE는 수백만 건의 데이터에서 어떻게 그렇게 빨리 조회할 수 있는지 알 수 있게 될 것이다.
  - 더불어 그동안 Sorted Set에 가지고 있던 의문들이 풀릴 것이다.
- **짚 리스트(ZIP LIST)**
  - 짚 리스트의 탄생 배경, 데이터 구조와 기본 동작,
  - Sorted Set에서 사용될 때 성능과 메모리를 얼마나 절약하는지를 알게 된다.

---

<< SSCAN

SORTED SETS Introduction

ZADD >>



redisgate@gmail.com



02.503.2235



서울시 강남구 강남대로 342 역삼빌딩 5층 (역삼동) 우 06242

Copyright © 2014-2024 redisGate  
All right reserved