redis

ZREMRANGEBYRANK key start stop

> **Available since 2.0.0.**
>
> **Time complexity:** O(log(N)+M) with N being the number of elements in the sorted set and M the number of elements removed by the operation.

Removes all elements in the sorted set stored at key with rank between `start` and `stop`. Both `start` and `stop` are `0` -based indexes with `0` being the element with the lowest score. These indexes can be negative numbers, where they indicate offsets starting at the element with the highest score. For example: `-1` is the element with the highest score, `-2` the element with the second highest score and so forth.

Return value

Integer reply: the number of elements removed.

Examples

```
redis> ZADD myzset 1 "one"
(integer) 1
redis> ZADD myzset 2 "two"
(integer) 1
redis> ZADD myzset 3 "three"
(integer) 1
redis> ZREMRANGEBYRANK myzset 0 1
(integer) 2
redis> ZRANGE myzset 0 -1 WITHSCORES
1) "three"
2) "3"
redis>
```

Related commands

ZADD
ZCARD
ZCOUNT
ZINCRBY
ZINTERSTORE
ZLEXCOUNT
ZRANGE
ZRANGEBYLEX
ZRANGEBYSCORE
ZRANK
ZREM
ZREMRANGEBYLEX
ZREMRANGEBYRANK
ZREMRANGEBYSCORE
ZREVRANGE
ZREVRANGEBYSCORE
ZREVRANK
ZSCAN
ZSCORE
ZUNIONSTORE

Join the discussion...

Ashwin Jayaprakash • 2 years ago

These offsets are confusing and they need a little experimentation before you can figure them out. So I thought I'd save others some trouble:

zadd myzset 1 "one"
zadd myzset 2 "two"
zadd myzset 3 "three"
zadd myzset 3 "tres"
zadd myzset 4 "four"
zadd myzset 4 "cuatro"

(Show all)
zrange myzset 0 -1 withscores

(Delete all)
zremrangebyrank myzset 0 -1

(Delete bottom 3) [Corrected to bottom from top on Dec 2, 2012]
zremrangebyrank myzset 0 2

(Delete top 3)
zremrangebyrank myzset -3 -1

4 ^ | v • Reply • Share ›

Petr Bela ➚ Ashwin Jayaprakash • a year ago

One more example:
(Delete all except the top 5) [safe to call even if there are less items, there is no "out of bounds" error]
zremrangebyrank myzset 0 -6

3 ^ | v • Reply • Share ›

Bonjour Tristesse ➚ Ashwin Jayaprakash • 2 years ago

Just a clarification.
To delete the 3 highest scores: zremrangebyrank myzset -3 -1
To delete the 3 lowest scores: zremrangebyrank myzset 0 2

1 ^ | v • Reply • Share ›

Ashwin Jayaprakash ➚ Bonjour Tristesse • 2 years ago

Corrected. Thanks.

^ | v • Reply • Share ›

Guest • 2 years ago

If I want to trim sorted set to keep the size under a limit, but need to remove entries with

least rank instead of highest rank. I cannot do that by ZRemRangeByRank without first get the actual size with ZCard. Is there any solution? Like ZRevRemRangeByRank?

2 • Reply • Share ›

Oasis Feng Guest • 2 years ago

I've done this with: ZRemRangeByRank {zset} 0 -{limit}-1

5 • Reply • Share ›

Jamesclarke40 • 2 years ago

Seriously, what is the point in this function. I'm looking for a method of returning all items in a set ranked but I can't seem to find it. This function seems so pointless by comparison?

1 • Reply • Share ›

Ben Cherry Jamesclarke40 • 2 years ago

Sounds like you're looking for the equivalent of SMEMBERS. Try `ZRANGE 0 -1` or `ZREVRANGE 0 -1` (ascending vs descending order).

• Reply • Share ›

Pieter Noordhuis Mod Jamesclarke40 • 2 years ago

ZRANGE can return all members in the set.

• Reply • Share ›

Yutong Zhao • 9 months ago

How can I get the identities of the removed elements?

• Reply • Share ›

Guest • 2 years ago

Is there any equivalent of ZRANGEBYRANK? It could useful to determine the score of a median, for example.

• Reply • Share ›

Roman Kashitsyn Guest • 2 years ago

Sorry, ZRANGE does exactly that.

• Reply • Share ›

Brantley Harris • 2 years ago

This is not actually remove range by rank because multiple items can have the same rank (if they have the same score, for instance). Rather should be ZREMRANGEBYINDEX (remove range by index). Higher scores are at the end of the range.

Example of limiting to the highest 20 items:

ZREMRANGEBYRANK myzset 0 -21

• Reply • Share ›

Pieter Noordhuis Mod ➦ Brantley Harris • 2 years ago

The ordering in sorted sets uses the lexicographical ordering of the member strings next to their scores. Since the members are unique by definition, no two score and member pairs will be the same.

When you retrieve the 20 highest pairs, you can always recompute their rank on the client side taking only the scores in account.

• Reply • Share ›

Antony Hatchkins ➦ Pieter Noordhuis • 2 years ago

Your second paragraph seems to be contradicting to the first one.

How can one correctly recompute the rank taking only the scores in account if both scores and member string is used to calculate the rank?

• Reply • Share ›

Brian Hammond ➦ Antony Hatchkins • a year ago

No, read the last part of the last sentence: "taking only scores in [sic] account". The idea is that if you so desire, you could display equal scores like this:

1. brian (10000)
joe (10000)
3. tom (9999)
etc

Note that there's no rank 2 here since brian and joe are tied for 1st.

• Reply • Share ›