**Title:** Redis ZREM

**Excerpt:** The ZREM command is used to remove one or more elements from a Redis sorted set stored at a given key. It returns the number of removed elements as the return value.

**Permalink:** redis-zrem

**Category:** Redis

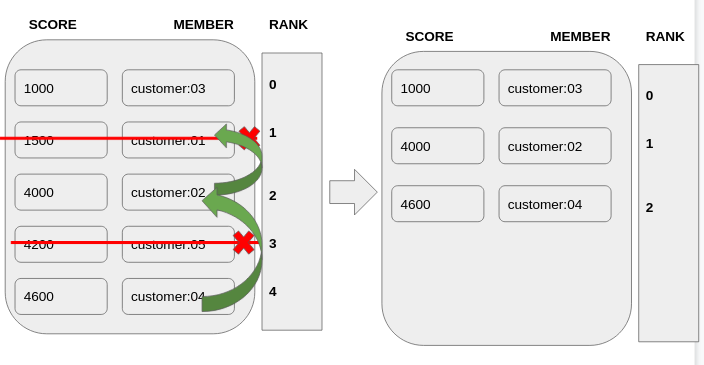
# 

# 

# 

# Redis Sorted Set Members

Redis sorted sets store unique elements in an ordered manner. Hence, it can be used in low latency applications such as priority queues, real-time leaderboards, secondary indexing, etc. Each of these set elements is assigned a rank and placed in ascending order based on the score value. Several commands are available to add, update, and remove the sorted set members efficiently. The most important thing about sorted sets is that after the addition or removal of a set member, it still manages to maintain the member ranks and order.



The above figure shows the Redis sorted set element removal scenario and how it behaves when multiple members are removed from it. It manages to keep the correct order and index values by placing each member accordingly. In the next section, we will be looking into the sorted set remove operation in a detailed manner.

# Removing Members with ZREM Command

The ZREM command is used to remove one or more members from a sorted set stored at a specified key. This command has logarithmic time complexity. If the command is used to remove one element, then the time complexity is proportional to the logarithmic value of the number of elements. If multiple elements have been specified to remove, then it will be O(number\_of\_elements\_to\_remove\* log(total\_number\_of\_set\_elements)).

## Syntax:

ZREM sorted\_set\_key member [member …]

***sorted\_set\_key***: This is the unique identifier where the sorted set is stored at.

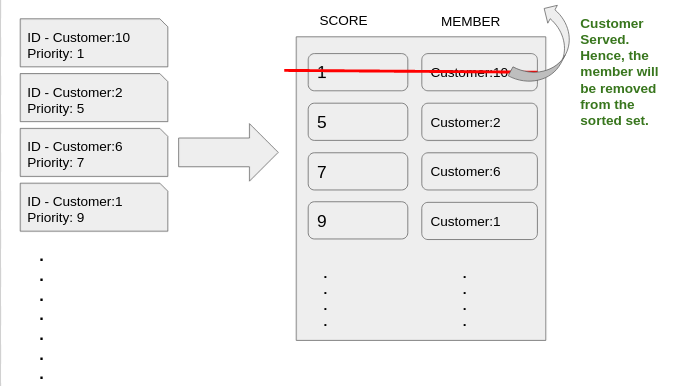
***member***: This is the member supposed to be removed.

The ZREM command will remove the specified members from the sorted set. It will return an integer reply which is the number of removed members. Whenever you specify a non-existing member, it will be skipped. Also, the command will throw an error if the given key doesn't hold a sorted set.

## Example - Manage Customer Order Queue with Redis ZREM

Let’s take a real-world scenario where a restaurant serves its customers based on the first-comers assigned with the highest priority. Upon serving the highest priority customers first, the restaurant should remove each customer entry from the queue. Redis sorted set will be an ideal data structure to implement this scenario in a memory-effective and efficient manner.

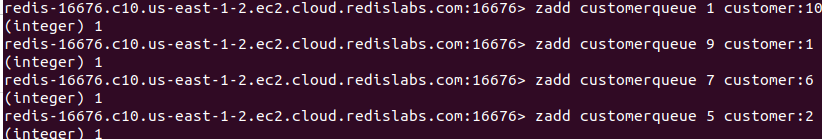
Each member of the Redis sorted set looks like the following.



As shown in the above illustration, each time a customer has been served by the restaurant, that member needs to be removed from the sorted set. We can use the ZREM command to achieve that.

Let’s add the four customers shown in the above figure. We will be using the ZADD command to create the sorted set ***customerqueue*** and add four members as shown in the following.

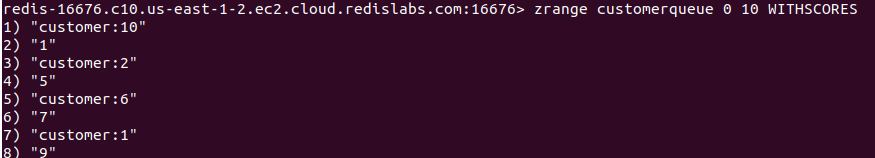
| **zadd** **customerqueue** 1 **customer**:10 **zadd** **customerqueue** 9 **customer**:1 **zadd** **customerqueue** 7 **customer**:6 **zadd** **customerqueue** 5 **customer**:2 |
| --- |



Let’s inspect the sorted set using the ZRANGE command.

| zrange customerqueue 0 10 WITHSCORES |
| --- |

Output:



As expected, the sorted set *customerqueue* has been ordered based on the priority value.

### Delete a Member from the Sorted Set

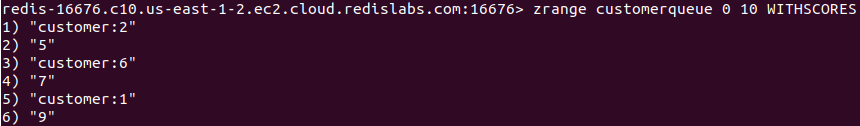
The *customer:10* who has the highest priority will be served first. Hence, the *customer:10* members has to be removed from the sorted set. Let’s use the ZREM command to remove the member.

| **ZREM** **customerqueue** **customer**:10 |
| --- |

Output:



As expected, the return value is 1 which means one entry has been removed. Let’s inspect the *customerqueue* again.



The member *customer:10* has been deleted successfully.

### Delete Multiple Members from the Sorted Set

Let’s say both *customer:2* and *customer:6* have been served from two windows parallelly. Hence, we should delete both the members from the sorted set. We can use the ZREM command to delete both the members at once.

| **ZREM** **customerqueue** **customer**:2 **customer**:6 |
| --- |

Output:



As expected, the return value is 2 which indicates two members have been deleted. Let’s inspect the whole sorted set again.



The ZREM command is recommended to use whenever you need to remove elements from a sorted set.

# Conclusion

To summarize, the ZREM command is used to remove one or multiple elements from a Redis sorted set stored at a given key. It simply deletes the specified members from the sorted set. Also, the order will be kept in the resulting sorted set. As stated above, it has a logarithmic time complexity which makes the command to be used in real-time applications. Hence, the ZREM command is recommended to use whenever you need to remove elements from a sorted set.