**Title:** Redis ZRANGEBYSCORE

**Excerpt:** The ZRANGEBYSCORE command is used to iterate over the members of a Redis sorted set stored at a given key. It can fetch members of a specified range of score values. Also, the command supports two optional arguments LIMIT and WITHSCORES which enhance the functionality furthermore.

**Permalink:** redis-zrangebyscore

**Category:** Redis

# 

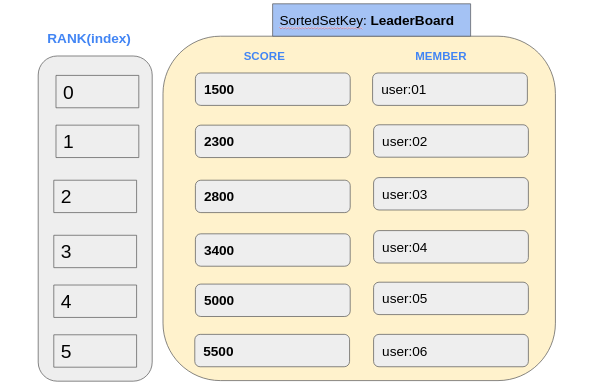
# 

# 

# Querying Redis Sorted Set Elements

Redis sorted sets are quite different from the normal sets. Even though both set types store unique members inside, the sorted sets put elements in an ordered manner. The main properties of the Redis sorted set are,

* Each element has been assigned to a unique index(rand) starting from 0
* Score order-based index per each element
* Lexicographical order-based index per each element



Hence, Redis sorted set members can be retrieved based on multiple measures such as score, rank(index), and lexicographical. ZRANGEBYSCORE and ZRANGE are the two main Redis sorted set commands that can be used to iterate over the members based on the previously stated measures. In this guide, we will focus on the ZRANGEBYSCORE command that is used to query members by a range of score values.

# The ZRANGEBYSCORE Command

Since the ZRANGEBYSCORE command has a logarithmic time complexity, it is very fast at retrieving sorted set elements. In addition, the command options like LIMIT reduce the retrieval time by a considerable amount. Hence, the ZRANGEBYSCORE command is safe to use in low latency real-time applications like online leaderboards, priority queues, and secondary indexing in general.

## Syntax:

| ZRANGEBYSCORE sorted\_set\_key minimum\_score maximum\_score [WITHSCORES] [LIMIT offset count] |
| --- |

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

***minimum\_score:*** The lower boundary score value of the specified range.

***maximum\_score:*** The higher boundary score value of the specified range.

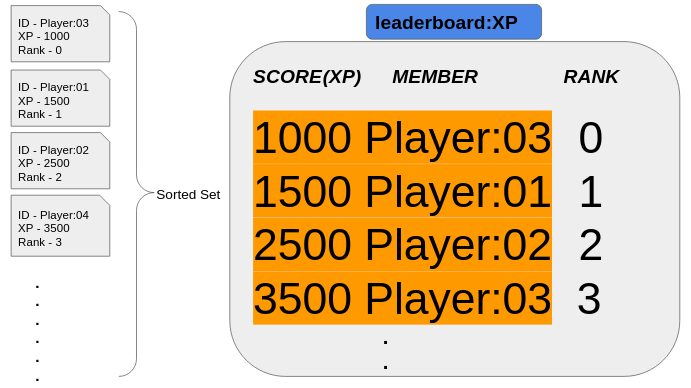
***WITHSCORES:*** This optional argument will return each element’s score

***LIMIT:*** This optional argument can be used to limit the returning element count from a specified position of the sorted set.

The ZRANGEBYSCORE command will return the sorted set members between the specified *minimum\_score* and *maximum\_score* scores. Those members will be returned in ascending order based on the score values. If the multiple members have the same scores, then the command will follow the lexicographical ordering.

## Example - Online Game Leaderboard Based on Player Experience

Let’s assume that it is an online game where each player can gain experience points when they complete missions, discover quests, and defeat enemies. Since Redis sorted sets are responsive and in-memory data structures where members are ordered based on a score, They can be used to store player details in turn the information can be returned with low latency.



As shown in the above illustration, the player information can be added to a Redis sorted set and manipulated later on when needed.

Let’s add the four players shown in the above illustration. We will be using the ZADD command to create and add player details to the sorted set *‘leaderboard:XP’.*

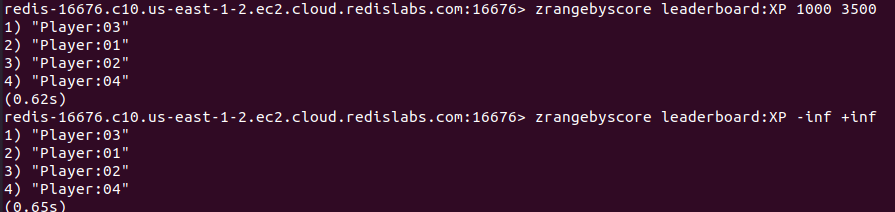
| **zadd** **leaderboard**:XP 1500 **Player**:01 **zadd** **leaderboard**:XP 2500 **Player**:02 **zadd** **leaderboard**:XP 1000 **Player**:03 **zadd** **leaderboard**:XP 3500 **Player**:04 |
| --- |

### Query All the Players in the Game Leaderboard

We can use the ZRANGEBYSCORE command to fetch all the players in the game leaderboard as shown in the following. The minimum and maximum score values have been specified as 1000 and 3500. We can use *-inf* and *+inf* values if we are not sure about the lowest and highest score values.

| **zrangebyscore** **leaderboard**:**XP** 1000 3500  OR  **zrangebyscore leaderboard:**XP-inf +inf |
| --- |

All the members will be returned as in the following output.



The output is sorted in ascending order by score values.

### Exclude the minimum\_score or maximum\_score values

In the above example, the min and max score values are inclusive of the range. We can exclude the min and max score values from the range by prefixing the score with **(** character as shown in the following.

| **zrangebyscore** **leaderboard**:XP (1000 (3500 |
| --- |

As shown in the following output, the output will exclude the members whose scores are 1000 and 3500.

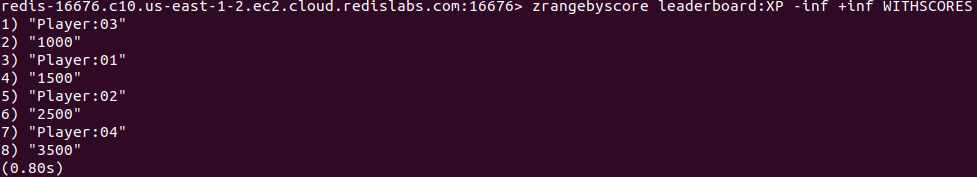


### Display Both the Member and Score Values Together

We can use the WITHSCORES optional argument with the ZRANGEBYSCORE command to display the score values per member.

| zrangebyscore leaderboard:XP -**inf** +**inf** WITHSCORES |
| --- |

Output:



### Limit the Number of Members Returned

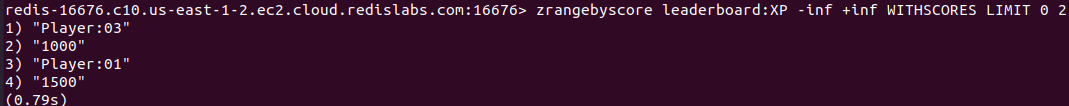
In some scenarios, we need to limit the number of returned members per call. Let’s say we need to get the top 2 members whose experience values(score) are the lowest. We can use the LIMIT argument with the count of 2 and offset of 0. The offset is the rank per member.

| zrangebyscore leaderboard:XP -**inf** +**inf** LIMIT 0 2 |
| --- |

This will return the top 2 members whose scores are the lowest.



You can use the LIMIT argument with the WITHSCORES argument as well.



## The ZRANGE Command Instead ZRANGBYSCORE

With the Redis 6.2.0 version, the ZRANGEBYSCORE command has been deprecated. Hence, we can use the ZRANGE command that behaves the same as ZRAGNEBYSCORE command when it is used with BYSCORE optional argument.

# Conclusion

In summary, the ZRANGEBYSCORE command is used to query the members between specified minimum and maximum score values of a Redis sorted set stored at a given key. As stated, this command has a logarithmic time complexity that can be used to iterate over a set of members with low latency. Also, it supports a couple of optional arguments LIMIT and WITHSCORES which limit the returned member count and display both the member-score value pairs.