**Title:** Redis ZDIFF

***Excerpt:*** The ZDIFF command is used to perform set differences between a given two sorted sets. It will produce another sorted set with all the members that only belong to the first set specified but not in the second set. This command takes the number of keys as the first argument and the WITHSCORES argument optionally.

**Permalink:** redis-zdiff

**Category:** Redis

# 

# 

# 

# Introduction to Set Difference

The set difference is one of the fundamental set operations that has always been on the list along with the other two set operations union and intersection. Furthermore, the difference between a given two sets produces another set. Consider the two sets A and B shown as follows.

A = {“Jack”, “Daniel”, “Mary”}

B = {“Daniel”, “Nick”, “Mary”, “Ricky”}

The set difference A-B can be written as follows.

A-B = {**“Jack”**, ~~“Daniel”~~, ~~“Mary”~~} - {~~“Daniel”~~, “Nick”, ~~“Mary”~~, “Ricky”}

Since the items “Daniel” and “Mary” are present in both sets, those will be ignored when taking the set difference A-B. Hence, the set difference A-B will be as follows.

A-B = {“Jack”}

It doesn’t need to be that A-B and B-A to be the same. Let’s consider the set difference B-A.

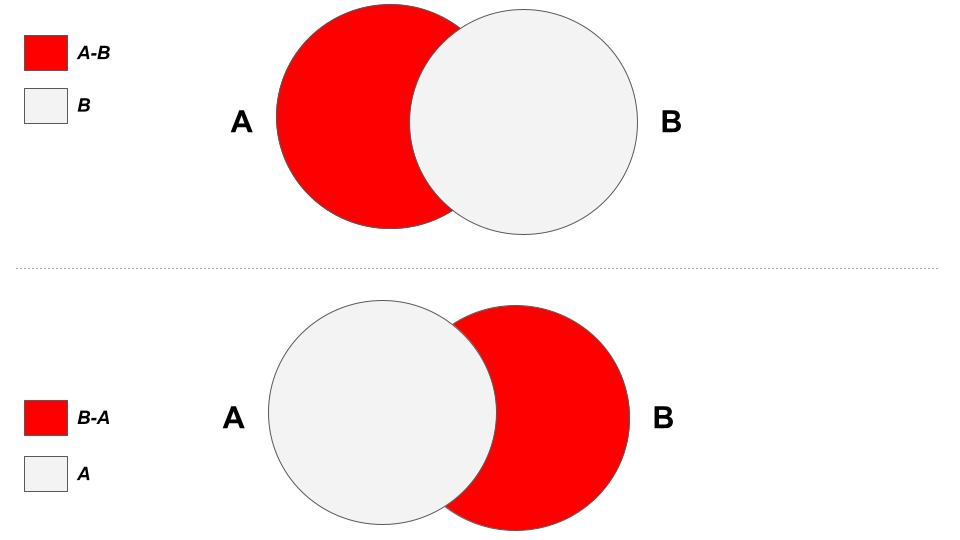
B = {“Daniel”, “Nick”, “Mary”, “Ricky”}

A = {“Jack”, “Daniel”, “Mary”}

B-A = {~~“Daniel”~~, **“Nick”**, ~~“Mary”~~, **“Ricky”**} - {“Jack”, ~~“Daniel”~~, ~~“Mary”~~}

B-A = {“Nick”, “Ricky”}

The set difference can be visualized using a Venn diagram in a more simple manner as follows.



# The ZDIFF Command

Redis sorted sets support set operations such as set union, intersection, and difference. In this guide, we will be focusing on the ZDIFF command that is used to produce the sorted set difference. The ZDIFF command returns another sorted set out of the provided sorted sets.

## Syntax:

ZDIFF command has a simple syntax where it is mandatory to specify the number of keys as the first argument as shown in the following.

| ZDIFF number\_of\_keys key [key...] [WITHSCORES] |
| --- |

***number\_of\_keys:*** The count of the keys of the specified sorted sets.

***key:*** A key of the specified sorted set.

***WITHSCORES:*** This is an optional parameter. When this argument is specified, scores of the returned set elements will be displayed.

This command returns a sorted set which is the difference between provided sorted sets. The element scores will be displayed if the *WITHSCORES* argument is passed.

## Use Case 01: Find Users who are Vegans

Let’s assume that a restaurant maintains two lists where one lists people who order vegetables and the other one records the people who order meat. Now they want to identify the vegans and send a vegetarian meal on their birthday. Redis sorted set data structure can be used in this scenario.

First, we will be creating a set to store people who ordered vegetable meals.

| zadd vegiconsumers 10 Jack  zadd vegiconsumers 5 Mary  zadd vegiconsumers 13 Ricky |
| --- |

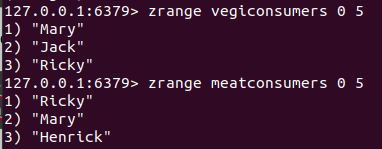
The score value has been used as the average number of meals ordered per month. Next, we will be creating another sorted set called *meatconsumers.*

| zadd meatconsumers 22 Henrick zadd meatconsumers 12 Mary zadd meatconsumers 5 Ricky |
| --- |

Let’s inspect the members of both the sorted sets using the ZRANGE command.

| zrange vegiconsumers 0 5 zrange meatconsumers 0 5 |
| --- |

Output:



As expected, both the sets have been populated with the specified members previously. Since the restaurant’s management team needs to query the vegans, we should populate the set difference between *vegiconsumers* and *meatconsumers*. The following command will populate a sorted set that consists of users who only ordered vegetable meals.

| zdiff 2 vegiconsumers meatconsumers |
| --- |

Since we take the set difference between two sets, the first argument is set to 2. Then, we have specified the keys of the two sets.

We can modify the above command to display the score value of the returned member.

| zdiff 2 vegiconsumers meatconsumers withscores |
| --- |

Output:



As expected, the set difference between the *vegiconsumers* and *meatconsumers* produces a sorted set with one member as shown in the following.

vegiconsumers = {Mary, Jack, Ricky}

meatconsumers = {Ricky, Mary, Henrick}

vegiconsumers - meatconsumers = {~~Mary~~, Jack, ~~Ricky~~} - {~~Ricky~~, ~~Mary~~, Henrick} = **{Jack}**

Hence, the only vegan customer is “Jack”.

## Use Case 02: Find Users who are Non-Vegans

Similarly, the restaurant management can query the non vegan customers. In that case, the set difference should be as follows.

***meatconsumers - vegiconsumers***

Let’s use the ZDIFF command with slight modification this time.

| zdiff 2 meatconsumers vegiconsumers withscores |
| --- |

Output:



As expected, the member “Henry” is the only customer who only ordered meals with meat without ordering any vegetable meal.

# Conclusion

In short, Redis supports fundamental set theory operations such as set union, intersection, and difference on the sorted set data type. The ZDIFF command is used to produce the difference between a given two sets. As mentioned, this command returns another sorted set which consists of the members that only belong to the first sorted set specified. As shown in the above use cases, it is mandatory to specify the number of keys as the first argument to this command. Overall, the ZDIFF command almost has logarithmic time complexity that can be used efficiently in low latency applications.