**Title:** Redis DECR

***Excerpt:*** The DECR command is used to decrement the integer value by one that is stored at a given key. This command only works with base-10 64-bit signed integer values. It can be used in counters which operate in constant time complexity.

**Permalink:** redis-decr

**Category:** Redis

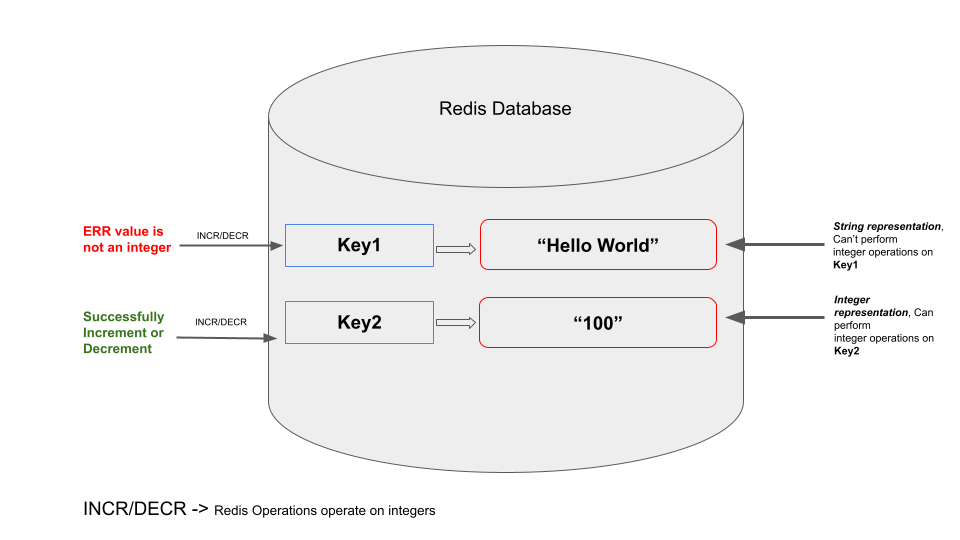
# 

# 

# 

# Redis Integer Manipulation

The string type is the most fundamental data type Redis offers. Redis strings are capable of storing texts, serialized JSON objects, images, audio, and even numerical values. In this guide, we will be focusing on numerical values and the manipulation of integers. Redis doesn’t support separate integer data types to hold numerical data. If a Redis key contains a string that can be represented as an integer, Redis can interpret number values as base-10 64-bit signed integers.



Number manipulation is an extremely useful feature to have in any database. So, Redis offers several useful operations to work on integer types such as INCR, DECR, INCRBY, etc. In this guide, the DECR command which operates on integer types will be explained in detail.

# The DECR Command

The DECR command is introduced to decrement a number value stored at a given key by one. This will work only on string values that can be represented as a base-10 64-bit integer. The DECR command will throw an error for non-string values or string values that can’t be represented as integers.

The following is the syntax of the DECR command.

| DECR key |
| --- |

***Key:*** The key which holds the string value.

The DECR command returns the resulting signed integer after the decrement operation. If the specified key doesn’t exist, the command will set the value to 0 and then decrement it by one. Also, an error will be thrown when the value is a non-string or can not be represented as a 64-bit signed integer. This command operates on constant time complexity(O(1)) which is extremely fast.

## Use Case - Decrement Player Health in an Online Game

Let’s assume an online game where each player’s health is incremented by one when a medi-pack is revealed. Similarly, the health decreases by one if the player falls or is hit by another player.

The ideal candidate to store a player’s health is a Redis string type where we can store the health as an integer. Let’s create a key *health:playerID:1* and set its value to 10. We can use the SET command as follows.

| set health:playerID:1 10 |
| --- |



We can use the GET command to verify the value stored at *health:playerID:1.*

| get health:playerID:1 |
| --- |



Let’s say the player with the ID 1, got a hit and the health should be decreased by one. So, the DECR command can be used as follows.

| decr health:playerID:1 |
| --- |

As expected, the DECR command prints the updated value after the decrement operation as shown in the following figure.



Also, we can inspect the value stored at key *health:playerID:1* with the GET command.

| get health:playerID:1 |
| --- |



As expected, the value has decreased by one.

The DECR operator is valid with negative numbers as well. Let’s store a new negative numerical value at the key *negative:value1*.

| set negative:value1 -3 |
| --- |

We can inspect the value using the GET command as follows.

| get negative:value1 |
| --- |



The following command shows how the DECR command is used on the key *negative:value1.*

| decr negative:value1 |
| --- |



### The DECR Command on Non-Integer Values

There are some edge cases when we try to use the DECR command on a Redis key that holds a non-string data type or a string that can’t be represented as an integer. The DECR command throws an error in those cases as shown in the following example.

| set notAnIntergerVal Hello |
| --- |

The “Hello” is a text that can’t be converted to a signed integer. So, the DECR command should raise an error if we call it against this key.

| decr notAnIntergerVal |
| --- |



### The DECR Command on Non-Existing Keys

In some cases, you might call the DECR command on a key that doesn’t exist in the Redis data store. So, the command will create the specified key and set its value to integer 0. At the same time, the value will be decreased by one.

| decr non-existing-key |
| --- |

As expected, the output is -1.



Overall, the DECR command is very useful to implement counters in low latency applications.

# Conclusion

In summary, DECR is one of the commands that operate on numerical values stored at a given Redis key. Redis doesn’t support a separate integer data type. So, the string type is used to hold numerical values as well. As examined, if the string value can be represented as a 64-bit signed integer, the DECR command can be used to decrement its value. Most importantly, it operates on constant time complexity. Also, the DECR command is widely used to implement counters.