**Title:** Redis HLEN

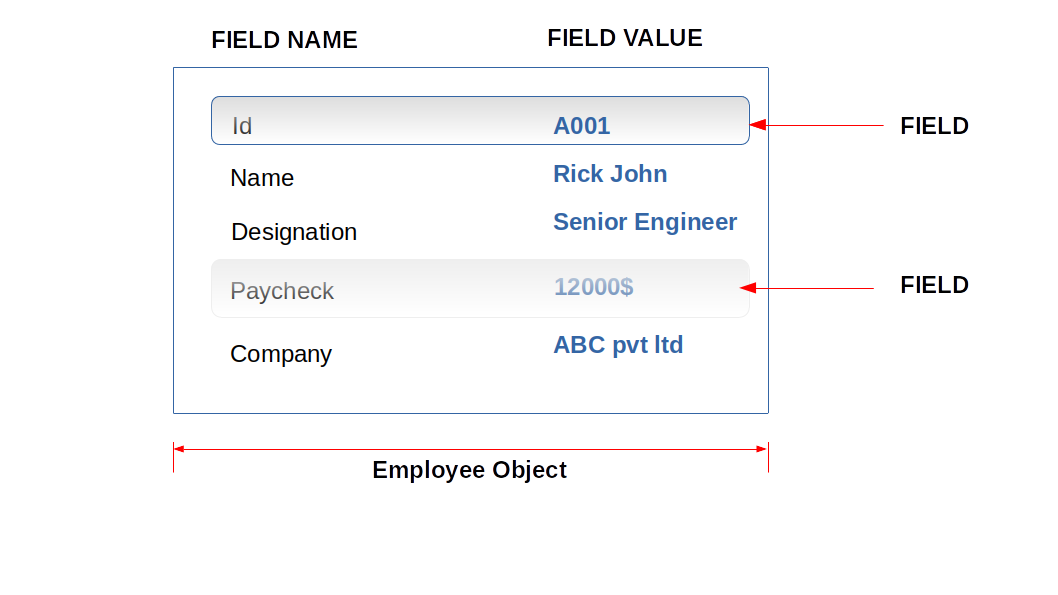
**Excerpt:** Redis hashes are one of the most popular data structures to use in Redis databases. Therefore, Redis supports different operations to manipulate hashes. The HLEN is one of many widely used commands to count the number of hash fields stored at a given Redis key. It operates with constant time complexity which is really good for high-performance applications.

**Permalink:** redis-hlen

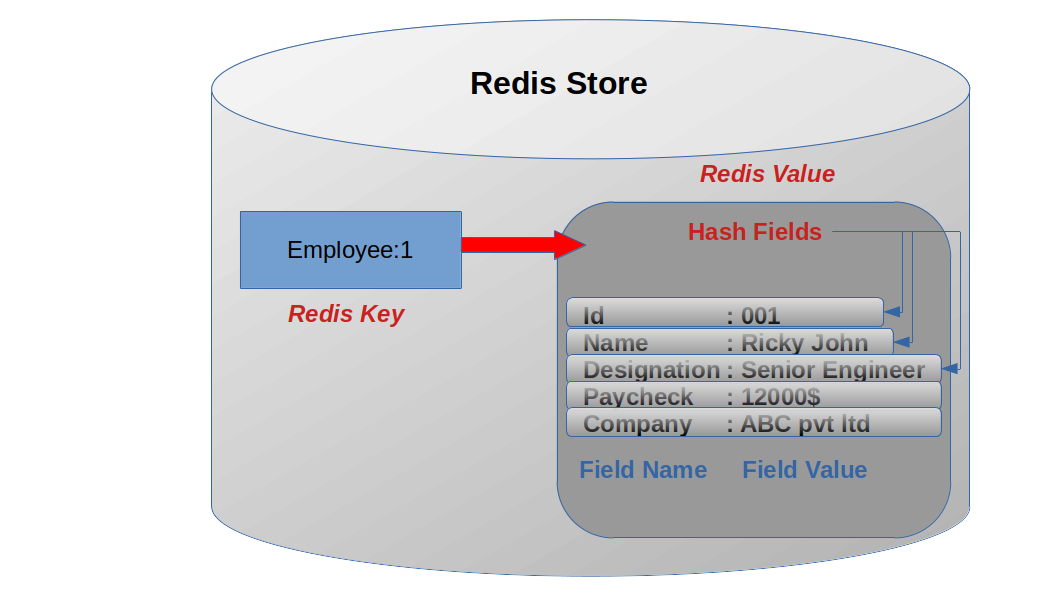
**Category:** Redis

# Redis Hash Fields

Hashes are a popular data type in Redis stores because they can comprehensively represent real-world objects. Real-world objects contain fields that are assignable to values as shown in the following.



In Redis, hashmap data structure provisions name-value pairs. One name-value pair can be identified as a field in a real-world entity. Let’s map the above Employee instance to a Redis hash.



In the above example, we got the top-level Redis key ‘*Employee:1*’ where we store our hash. There are four hash fields representing four attributes of the Employee object. Redis hashes can store more than 4 billion hash fields which are more than enough to represent your objects. The best thing is it still takes minimal space in your Redis database instance. It is more efficient to maintain 1000 hash fields within 10 top-level Redis keys to store 10000 key-value data than storing that amount of data in 10000 top-level Redis keys.

# Redis HLEN command

In some applications, it is necessary to count the number of properties or fields available for a given object. If it is stored in a Redis hash, then it is one command away from retrieving the number of hash fields at the specified key. The ‘**HLEN’** command can be used to count the number of hash fields contained in a given hash.

## Syntax

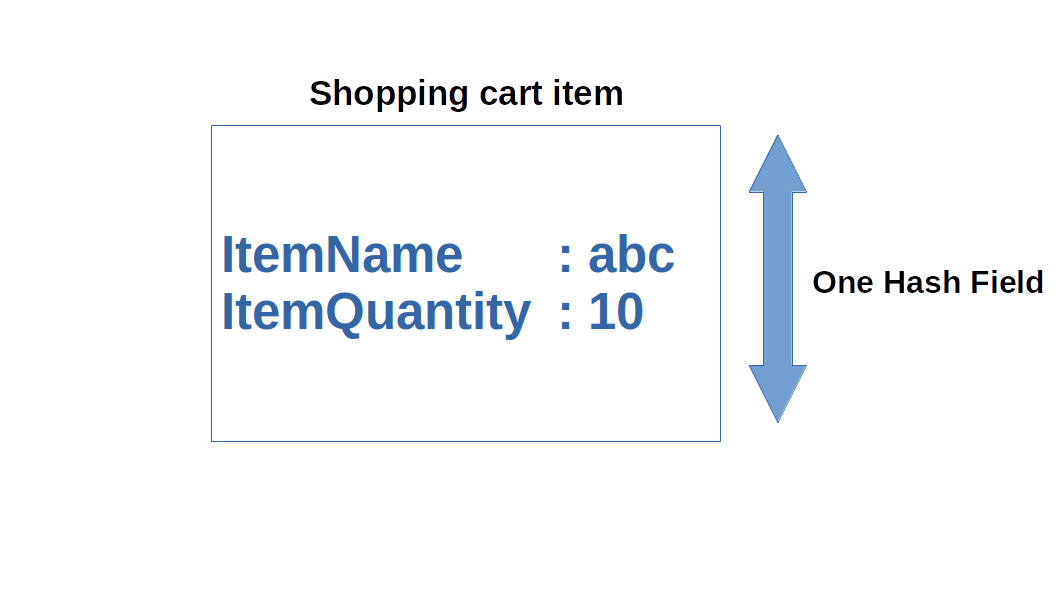
| HLEN hash\_key |
| --- |

**hash\_key**: This is the top-level Redis key that the hash has been stored at.

This command returns an integer value which is the number of hash fields stored at a given key. When the hash\_key doesn't exist, this command will return 0. This command has constant time complexity O(1) which motivates you to use it in high-performance applications.

## Example - Counting number of items in a shopping cart

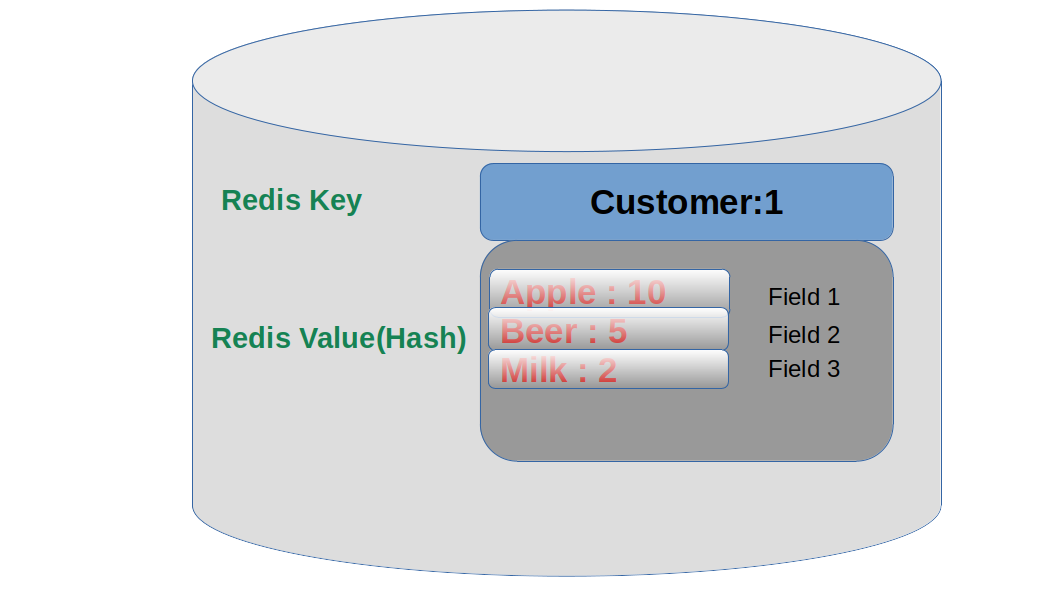
Let’s assume that we are using the Redis database to store shopping cart items. The idea is to identify each customer with a Redis key. Each Redis key stores multiple hash fields that represent a shopping cart item. The shopping cart item object would look like the following.



We can use the HSET command to add some cart items for ‘*customer:1’*.

| HSET customer:1 apple 10 beer 5 milk 2 |
| --- |

Now the *customer:1* object should look like the following.



There are 3 fields stored at the ‘customer:1’ key. We will be using the HLEN command to count the number of fields at the ‘customer:1’ key.

| **HLEN** **customer**:1 |
| --- |

Output:



The output is 3 as expected.

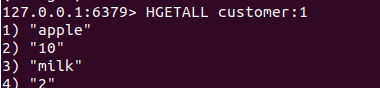
Let’s delete the ‘beer’ field at the *customer:1* key. The HDEL command can be used to delete a field from a given key.

| **HDEL** **customer**:1 **beer** |
| --- |

Now we can check the existing hash fields by running the HGETALL command.

| **HGETALL** **customer**:1 |
| --- |

Output:

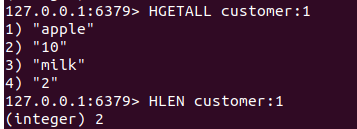


The ‘beer:5’ field has been successfully deleted. Let’s count the number of fields again using the HLEN command.

| **HLEN** **customer**:1 |
| --- |

Since we got only two hash fields left at the key ‘customer:1’, the output should be 2.

Output:



It is recommended to use the HLEN command when there is a requirement to count the number of hash fields stored at a given key. It is a very fast and easy command to use in your Redis based applications.

Let’s check the output for non-existence key as well.

| HLEN non-existence-key |
| --- |

Output:



The output is 0 as expected.

# Conclusion

Redis can store different types of data structures for a specific key. The hashes are one of the most popular data structures used in Redis databases. Hashes can be used to represent and manipulate simple objects and data effectively. Redis supports several hash operations. The HLEN can be used to count the number of hash fields stored at a given Redis key. Usually, it returns an integer that is the count of the hash fields. If the Redis key doesn’t exist, it will return 0. It is really fast since we do not need to perform a separate READ operation.