Redis - Data Types

Redis supports 5 types of data types.

Strings

Redis string is a sequence of bytes. Strings in Redis are binary safe, meaning they have a known length not determined by any special terminating characters. Thus, you can store anything up to 512 megabytes in one string.

Example

```
redis 127.0.0.1:6379> SET name "tutorialspoint"
OK
redis 127.0.0.1:6379> GET name
"tutorialspoint"
```

In the above example, **SET** and **GET** are Redis commands, **name** is the key used in Redis and **tutorialspoint** is the string value that is stored in Redis.

Note − A string value can be at max 512 megabytes in length.

Hashes

A Redis hash is a collection of key value pairs. Redis Hashes are maps between string fields and string values. Hence, they are used to represent objects.

Example

```
redis 127.0.0.1:6379> HMSET user:1 username tutorialspoint password
tutorialspoint points 200
OK
redis 127.0.0.1:6379> HGETALL user:1
1) "username"
2) "tutorialspoint"
3) "password"
4) "tutorialspoint"
```

```
5) "points"
6) "200"
```

In the above example, hash data type is used to store the user's object which contains basic information of the user. Here **HMSET, HGETALL** are commands for Redis, while **user − 1** is the key.

Every hash can store up to 2^{32} - 1 field-value pairs (more than 4 billion).

Lists

Redis Lists are simply lists of strings, sorted by insertion order. You can add elements to a Redis List on the head or on the tail.

Example

```
redis 127.0.0.1:6379> lpush tutoriallist redis
(integer) 1
redis 127.0.0.1:6379> lpush tutoriallist mongodb
(integer) 2
redis 127.0.0.1:6379> lpush tutoriallist rabitmq
(integer) 3
redis 127.0.0.1:6379> lrange tutoriallist 0 10

1) "rabitmq"
2) "mongodb"
3) "redis"
```

The max length of a list is 2^{32} - 1 elements (4294967295, more than 4 billion of elements per list).

Sets

Redis Sets are an unordered collection of strings. In Redis, you can add, remove, and test for the existence of members in O(1) time complexity.

Example

```
redis 127.0.0.1:6379> sadd tutoriallist redis
(integer) 1
```

```
redis 127.0.0.1:6379> sadd tutoriallist mongodb
(integer) 1
redis 127.0.0.1:6379> sadd tutoriallist rabitmq
(integer) 1
redis 127.0.0.1:6379> sadd tutoriallist rabitmq
(integer) 0
redis 127.0.0.1:6379> smembers tutoriallist

1) "rabitmq"
2) "mongodb"
3) "redis"
```

Note − In the above example, **rabitmq** is added twice, however due to unique property of the set, it is added only once.

The max number of members in a set is 2^{32} - 1 (4294967295, more than 4 billion of members per set).

Sorted Sets

Redis Sorted Sets are similar to Redis Sets, non-repeating collections of Strings. The difference is, every member of a Sorted Set is associated with a score, that is used in order to take the sorted set ordered, from the smallest to the greatest score. While members are unique, the scores may be repeated.

Example

```
redis 127.0.0.1:6379> zadd tutoriallist 0 redis
(integer) 1
redis 127.0.0.1:6379> zadd tutoriallist 0 mongodb
(integer) 1
redis 127.0.0.1:6379> zadd tutoriallist 0 rabitmq
(integer) 1
redis 127.0.0.1:6379> zadd tutoriallist 0 rabitmq
(integer) 0
redis 127.0.0.1:6379> ZRANGEBYSCORE tutoriallist 0 1000

1) "redis"
2) "mongodb"
3) "rabitmq"
```