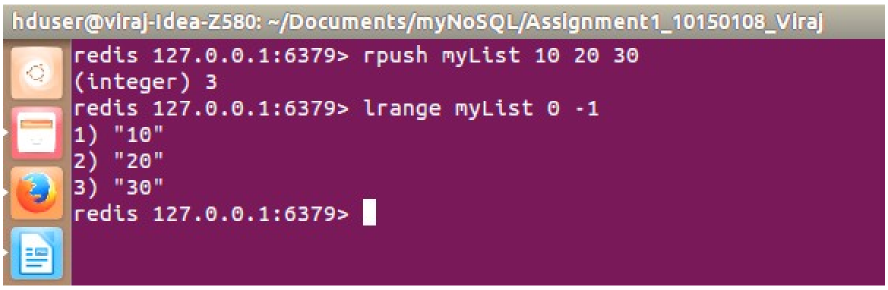
Assignments for REDIS

Viraj (10150108)

1. Create a redis list with value [10, 20, 30]

> rpush myList 10 20 30
> lrange myList 0 -1

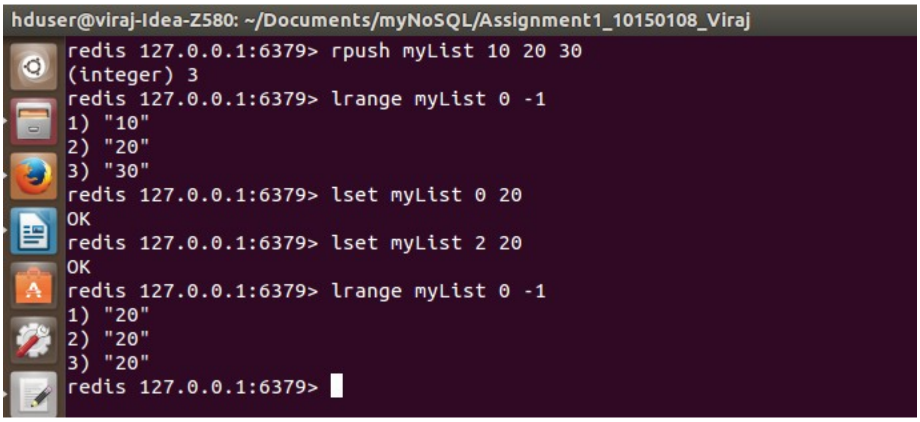


hduser@viraj-Idea-Z580: ~/Documents/myNoSQL/Assignment1_10150108_Viraj

```
redis 127.0.0.1:6379> rpush myList 10 20 30
(integer) 3
redis 127.0.0.1:6379> lrange myList 0 -1
1) "10"
2) "20"
3) "30"
redis 127.0.0.1:6379>
```

2. Change the list to [20, 20, 20]

> lset myList 0 20   <<< 0 is the index position of the list
> lset myList 2 20   <<< 2 is the index position of the list
> lrange myList 0 -1

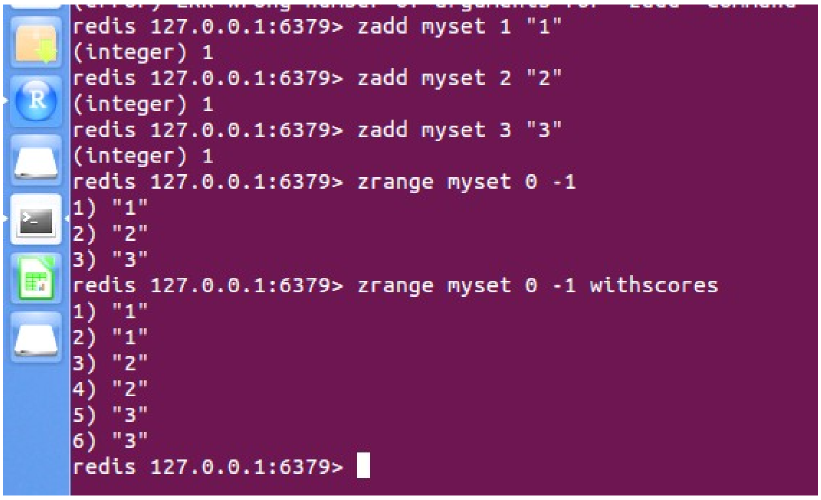


hduser@viraj-Idea-Z580: ~/Documents/myNoSQL/Assignment1_10150108_Viraj

```
redis 127.0.0.1:6379> rpush myList 10 20 30
(integer) 3
redis 127.0.0.1:6379> lrange myList 0 -1
1) "10"
2) "20"
3) "30"
redis 127.0.0.1:6379> lset myList 0 20
OK
redis 127.0.0.1:6379> lset myList 2 20
OK
redis 127.0.0.1:6379> lrange myList 0 -1
1) "20"
2) "20"
3) "20"
redis 127.0.0.1:6379>
```

3. Create a sorted set with the value {1, 2, 3 }

> zadd myset 1 “1”
> zadd myset 2 “2”
> zadd myset 3 “3”
> zrange myset 0 -1
> zrange myset 0 -1 withscores



```
redis 127.0.0.1:6379> zadd myset 1 "1"
(integer) 1
redis 127.0.0.1:6379> zadd myset 2 "2"
(integer) 1
redis 127.0.0.1:6379> zadd myset 3 "3"
(integer) 1
redis 127.0.0.1:6379> zrange myset 0 -1
1) "1"
2) "2"
3) "3"
redis 127.0.0.1:6379> zrange myset 0 -1 withscores
1) "1"
2) "1"
3) "2"
4) "2"
5) "3"
6) "3"
redis 127.0.0.1:6379>
```

4. Add {3, 4, -5} to this set
Display the list contents and the ordered set contents

<u>Using “Z” commands</u>

In a sorted set , we cannot have duplicate members even for different scores, so when we try to add 3 in the existing set the score gets updated for the same. Also if the score is not properly mentioned the sorting goes for a toss.

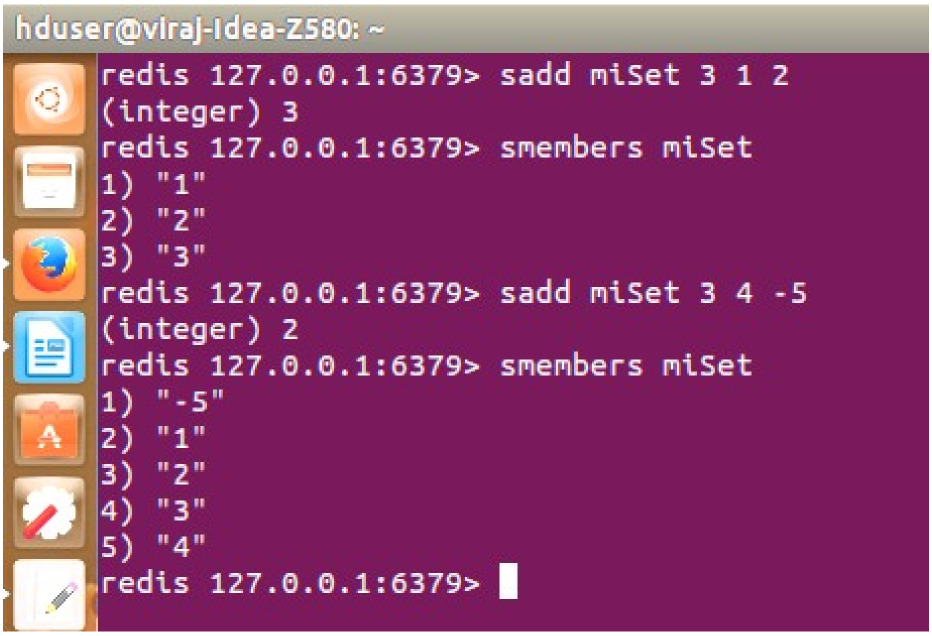
So I am adding 3, 4, 5 to the existing “**myset**” with scores as “4”,“5”,“0” respectively.
Due to this, the existing 3 which was having a score of “3” got updated to “4” and as I need a final sorted set, I am giving a score of 0 to “-5”

> zadd myset **4 3 5 4 0 -5**

<u>Using “S” commands</u>

Using “SADD” we can simply pass the members/elements and it gets sorted out based on the values.

```
> sadd miSet 3 1 2
> smembers miSet
> sadd miSet 3 4 -5
> smembers miSet
```



```
redis 127.0.0.1:6379> sadd miSet 3 1 2
(integer) 3
redis 127.0.0.1:6379> smembers miSet
1) "1"
2) "2"
3) "3"
redis 127.0.0.1:6379> sadd miSet 3 4 -5
(integer) 2
redis 127.0.0.1:6379> smembers miSet
1) "-5"
2) "1"
3) "2"
4) "3"
5) "4"
redis 127.0.0.1:6379>
```