Non-Relational Databases

Redis 2

Prof. Dr. Jürgen Heym

Hochschule Hof

3 – Redis – CRUD & Datetypes

1. Open the Redis-CLI and check the connection to your Redis-Server using the command ping.

```
$ redis-cli
redis 127.0.0.1:6379> ping
```

2. Check out the redis help command.

```
redis...> help
```

3. Insert a key-value-pair into your redis database. Choose a key and set the value to the URL of Hof University. All keys should start with your initials (see example). Retrieve the value of your new key.

```
redis...> SET jh_hshof http://www.hof-university.de
redis...> GET jh_hshof
```

3 – Redis – CRUD & Datetypes

4. Insert several key-value-pairs into your redis database using the MSET command. Retrieve several values using the MGET command.

```
redis...> MSET jh_unibt http://www.uni-bayreuth.de     \
               jh_hsco https://www.hs-coburg.de/   ...

redis...> MGET jh_unibt jh_hsco ...
```

6. Set key-value-pairs representing an integer number. Check arithmethic operations like INCR, INCRBY, INCRBYFLOAT, DECR and DECRBY.

```
redis...> SET jh_anzahl 3
redis...> INCR jh_anzahl
redis...> GET jh_anzahl
```

3 – Redis – CRUD & Datetypes

6. Transactions

```
redis...> MULTI
redis...> SET jh_tum http://www.tum.de
redis...> INCR jh_anzahl
redis...> EXEC
```

There is not a real rollback! With command DISCARD one can stop a transaction.

7. Hashes may hold an unlimited number of key-value-pairs. User profile data may be stored as a hash, e.g. user:jheym.

```
redis...> HMSET user:jheym name "Juergen Heym" password "s3cret"
redis...> HKEYS user:jheym
redis...> HVALS user:jheym
redis...> HGET user:jheym name
```

Hashes may not be nested!

3 – Redis – CRUD & Datetypes

8. Lists contain one or several ordered values and may be used as FIFO queues (first in first out) or as LIFO queues (last in first out). Construct a list that holds the website favorites of a user.

```
redis...> RPUSH jheym:favorites jh_hshof jh_hsco jh_unibt
```

The length of a list is returned by command LLEN.

```
redis...> LLEN jheym:favorites
```

Command LRANGE is used to get a range of list elements.

```
redis...> LRANGE jheym:favorites 0 0
```

Try out the following list commands: LSET, LREM, LTRIM, LINDEX, LINSERT.

3 – Redis – CRUD & Datetypes

9. More lists …

   You can add and remove lists values left and right:

```
redis...> LPOP jheym:favorites
redis...> RPOP jheym:favorites
redis...> LPUSH jheym:favorites jh_unimr
redis...> RPUSH jheym:favorites jh_unihh
```

   You can remove from one list and add to another one:

```
redis...> RPOPLPUSH todo done
```

   There is now variant for this command like e.g. LPOPRPUSH!

3 – Redis – CRUD & Datetypes

10. Blocking lists

We want to implement a real time comment system. Website visitors will comment the website with likes. Several clients may write comments in a queue. Another client shall check the comments. For this purpose we need blocking list commands.

```
redis...> BRPOP comments 300
```

BRPOP waits max. 5 min for new values in list „comments“ and blocks max this amount of time.

Open another console window with a new redis-cli and add a comment:

```
redis...> LPUSH comments "Tough website ..."
```

Control the result in your first console window!

3 – Redis – CRUD & Datetypes

11. Sets

Sets are unordered collections without duplicates. You can applay set operations on sets.

```
redis...> SADD jh_mag_1 frankenpost.de colorfoto.de
redis...> SMEMBERS jh_mag_1
```

Create another set with magazines.

```
redis...> SADD jh_mag_2 fotomagazin.de colorfoto.de
```

Create the union, difference and intersection of the two sets.

```
redis...> SINTER jh_mag_1 jh_mag_2
redis...> SDIFF  jh_mag_1 jh_mag_2
redis...> SUNION jh_mag_1 jh_mag_2
```

3 – Redis – CRUD & Datetypes

12. More sets …

Union, difference and intersection may be stored in new sets:

```
redis...> SUNIONSTORE jh_printmedien jh_zeitschriften jh_magazine
```

Move elements of a set to another set:

```
redis...> SMOVE ...
```

Determine the cardinality of a set:

```
redis...> SCARD ...
```

Remove elements of a set:

```
redis...> SREM ...
```

Sets are unsorted, so don‘t expect left-right-variants for set commands.

3 – Redis – CRUD & Datetypes

13. Sorted sets

Sorted sets are sorted like lists, unique like sets and use key-value-pairs like hashes, but numerical weights determine the positions of the elements. This is used for prioirised queues with random access.

Store the rank of visited websites as sorted set:

```
redis...> ZADD visits 543 jh_hshof 899 jh_unibt 400 jh_hsco
```

The priority is given by the interger counter. You can increment and decrement these priorities directly:

```
redis...> ZINCRBY visits   1 jh_hshof      ==> ergibt 544
redis...> ZINCRBY visits  -4 jh_hshof      ==> ergibt 540
redis...> ZADD visits 545 jh_hshof ==> ergibt 545
```

3 – Redis – CRUD & Datetypes

14. More sorted sets …

You may show the content of sorted sets in different ways:

a) Show the first three elements in ascending order:

```
redis...> ZRANGE visits 0 2
```

b) Show all elements in ascending order with priority values (scores):

```
redis...> ZRANGE visits 0 -1 withscores
```

c) Same but reverse sorting:

```
redis...> ZREVRANGE visits 0 -1 withscores
```

d) Show all elements in ascending order in range 500 < score <= 800:

```
redis...> ZRANGEBYSCORE visits (500 800 withscores
```

e) Show all elements in ascending order with priority values (scores) in the range - ∞ (-inf) to + ∞ (inf).

```
redis...> ZRANGEBYSCORE visits -inf inf
```

3 – Redis – CRUD & Datetypes

15. More sorted sets …

    There are different possibilities to remove elements from a sorted set. Check them out!

```
redis...> ZREMRANGEBYRANK ...
redis...> ZREMRANGEBYSCORE ...
```

16. Build the union of two sorted sets.

    You have to pay attention on how to take care of the weights. There are min, max and sum stratgies.

```
redis...> ZUNIONSTORE newSet setCount set1 set2
                      [WEIGHTS g1[ g2 ...]]        \
                      [AGGREGATE SUM|MIN|MAX]
```

Literatur

- Redis Essentials (M. da Silva, H. Lopez Tavarez)
  ISBN-13: 978-1784392451
- Mastering Redis (J. Nelson)
  ISBN-13: 978-1783988181
- Sieben Wochen, sieben Datenbanken
  Moderne Datenbanken und die NoSQL-Bewegung
  E. Redmon & J. R. Wilson, Oreilly®
  ISBN 978-3-86899-791-0
- Redis
  https://redis.io/