

Databases and information systems laboratory

CS313

IIT Dharwad

Handout 9
18 – 10 – 2023

The following set-up is Key-value pair Database. Use *Redis* to do the following¹. Submit your answers in a text file. (★) denotes questions that you can try on your own.

Server-Client setup

1. Open a *Redis server* using the command `redis-server`
2. Open a *Redis client* using the command `redis-cli`
3. Verify that the client is connected to the server using the command `PING`
4. Print *Hello World* on the client terminal using the command `ECHO`

Insert and Delete

1. Create a key *client1* whose value is *Alice* using the command `SET`
2. Find the value for the key *client1* using the command `GET`
3. (★) Try to insert a new value with *client1*. What is the result?
4. (★) Try to find the value of a key that is not present in the database. What is the result?
5. Remove the key-value pair with the key *client1* using the command `DEL`

¹Install *Redis* using the link <https://redis.io/docs/getting-started/>

6. (★) Try to delete a key-value pair where the key is not in the database.
What is the result?
7. (★) Create a key *client:1:name* whose value is *Alice*
8. (★) Create a key *client:1:level* whose value is 1

Update values

1. Increment the value whose key is *client:1:level* using the command **INCR**
2. Decrement the value whose key is *client:1:level* using the command **DECR**

Delete based on time

1. Check if the key *client:1:name* exists, using the command **EXISTS**
2. (★) Check if the key *client:1:address* exists.
3. (★) Add new key *Message* with value *Hello world*
4. Write a command to delete the key *Message* with its value after 50 seconds using the command **EXPIRE**
5. Check the remaining time (in seconds) for the key *Message* using the command **TTL**
6. (★) Check the remaining time (in seconds) for a key that has not been assigned an expiry time.
7. (★) Check the remaining time (in seconds) for a key that does not exist in the database.
8. Add a new key *Message2* with value *Hello universe* and also set its expiration to 150 seconds using the command **SETEX**
9. Stop the expiration time for the key *Message2* using the command **PERSIST**
10. (★) Try to make a key persistent that does not have an expiration time.
What is the result?
11. (★) Try to make a key persistent that does not exist in the database.
What is the result?

Inserting multiple keys together

1. Add the keys *server:1:name* and *server:1:port* with values *Apache* and 8000 respectively, using the command **MSET**
2. Append *lite* to the value of the key *server:1:name* using the command **APPEND**
3. Rename the key *server:1:name* to *server:1:identifier* using the command **RENAME**
4. Delete everything from the database using the command **flushall**

List

1. Create a key *Customers* whose value is a **list** containing *Alice*, *Bob*, *Charles* using the command **RPUSH**
2. (★) remove the key *Customer* and repeat the previous question with **LPUSH**
3. Find all the names in the list with the key *Customer* using the command **LRANGE**
4. (★) Find the first 2 names in the list with the key *Customer*
5. Add a new name *Harry* to the list with the key *Customer* after Bob using the command **LINSERT ... AFTER**
6. (★) Add a new name *Dumbledore* to the list with the key *Customer* before Bob using the command **LINSERT ... BEFORE**
7. Find the length of the list for the key *Customer* using the command **LLEN**
8. Remove the right most element of the list in the key *Customer* using the command **RPOP**
9. (★) Remove the left most element of the list in the key *Customer* using the command **LPOP**
10. (★) Delete everything from the database.

Set

1. Create a key *courses* whose values form a set *CS*, *Maths*, *Physics* using the command **SADD**
2. (★) Add *Economics* to the set whose key is *courses*
3. Display all the members of the set whose key is *courses* using the command **SMEMBERS**
4. (★) Again add *Physics* to the set whose key is *courses*. Does it work?
5. Display the **size** of the set whose key is *courses* using the command **SCARD**
6. Check if *CS* is a member of the set of the value *courses* using the command **SISMEMBER**
7. (★) Check if *Chemistry* is a member of the set of the value *courses*
8. Move the element *Physics* from the key *courses* to *BasicScience* using the command **SMOVE**
9. Delete the element *CS* from the key *courses* using the command **SREM**
10. (★) Delete the element *Chemistry* from the key *courses*. Does it work?
11. (★) Delete everything from the database.

Ordered set

1. Create a new key *Clients* whose value is a sorted list that contains client names and their income as the *score*. Use the command **ZADD** and the values *(Harry, 3500)* *(Alice, 3300)* *(Charles, 3500)* *(Robert, 3400)*
2. Display all the names in the sorted list with the key *Clients* using the command **ZRANGE**
3. (★) Add *(Charles, 3200)* to the ordered list with key *Clients*. What is the result?
4. Find the **position** of *Charles* in the key *Clients* using the command **ZRANK**.
5. (★) Find the position of *Dumbledore* in the key *Clients*. What is the result?

6. Increase the salary of Robert by 100 using the command **ZINCRBY**
7. (★) Check the new positions of all the elements.
8. (★) Delete everything from the database.

Hash set

1. Create a *Hash set* called *Customer:1* with two keys *name*, *income* with values *Alice*, *3200* respectively. Use the command **HSET**
2. Find the value of *name* for *Customer:1* using the command **HGET**
3. (★) Find the value of *address* for *Customer:1*
4. (★) Find the value of *name* for *Customer:2*
5. Find the value of all keys for *Customer:1* using the command **HGETALL**
6. (★) Find the value of all keys for *Customer:2* using the command **HGETALL**
7. Find all the keys for *Customer:1* using the command **HKEYS**
8. Find all the values for *Customer:1* using the command **HVALS**
9. (★) Find all the keys for *Customer:2*
10. (★) Find all the values for *Customer:2*
11. Display the number of keys in *Customer:1* using the command **HLEN**
12. Decrement the value of *income* for *Customer:1* using the command **HINCRBY**
13. Delete the key *income* and its value for *Customer:1* using the command **HDEL**

Save and Exit

1. Save the current database onto the disk using the command **SAVE**
2. Close the client connection using the command **QUIT**