Redis

5 Data type에 대한 이해

Data Type

Key/Value

Data Type

Key/Value

binary and **text**
ex) “name”, “123”, “#!:()”

Data Type

Key/Value

Data Type

Strings
Lists
Sets
Sorted sets
Hashes
Geospatial
Bitmap

Data Type

Strings

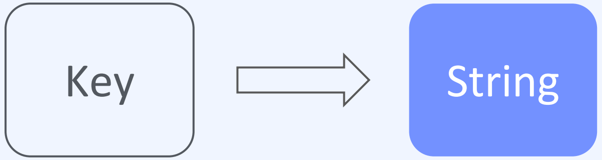
Strings

Data Type

Strings

Strings

- **대표 기본 타입**으로 바이너리, 문자 데이터를 저장
  - maximum 512MB
- 증가 감소에 대한 원자적 연산
  - increment/decrement



Data Type

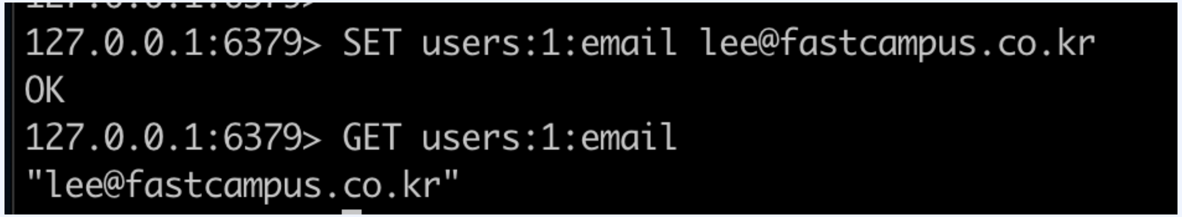
Strings

command

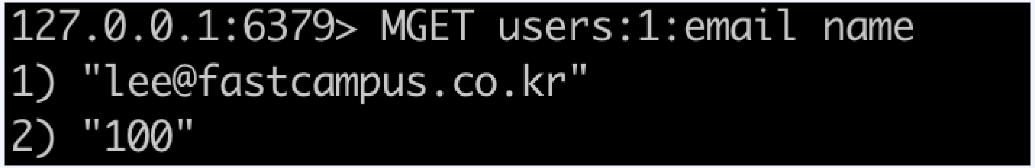
- SET
- SETNX
- GET
- MGET
- INC
- DEC

Data Type

Strings example



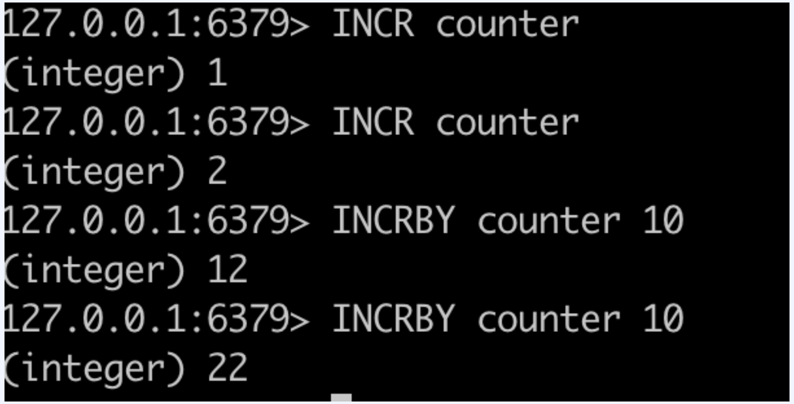
```
127.0.0.1:6379> SET users:1:email lee@fastcampus.co.kr
OK
127.0.0.1:6379> GET users:1:email
"lee@fastcampus.co.kr"
```



```
127.0.0.1:6379> MGET users:1:email name
1) "lee@fastcampus.co.kr"
2) "100"
```

Data Type

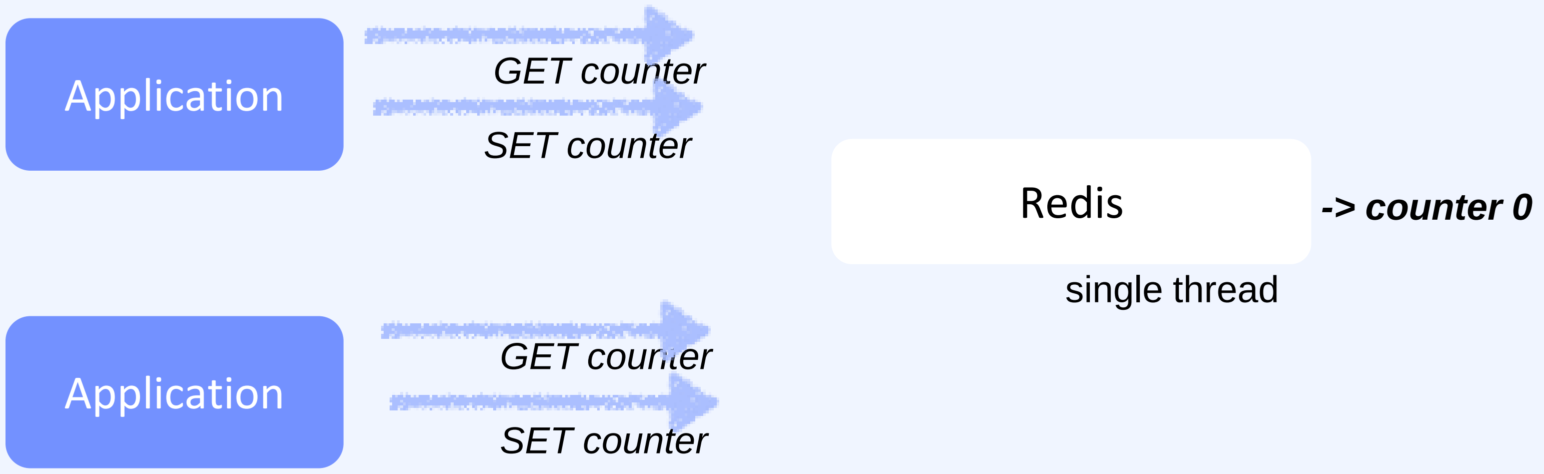
Strings example



```
127.0.0.1:6379> INCR counter
(integer) 1
127.0.0.1:6379> INCR counter
(integer) 2
127.0.0.1:6379> INCRBY counter 10
(integer) 12
127.0.0.1:6379> INCRBY counter 10
(integer) 22
```

Data Type

Strings example



Key 주요 명령어

잠깐, Key 명령어를 알아 봅시다

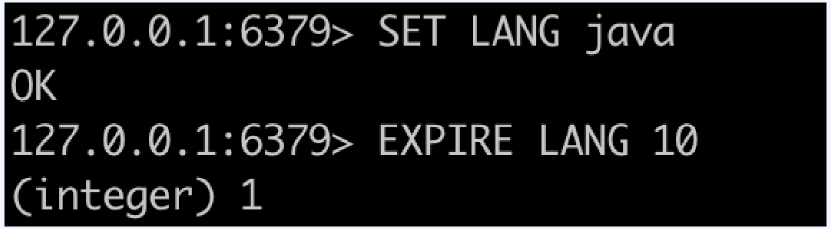
Data Type

Key 주요 명령어

TTL(Time To Live)

- EXPIRE [KEY] [SECOND]
- TTL [KEY]

TTL command

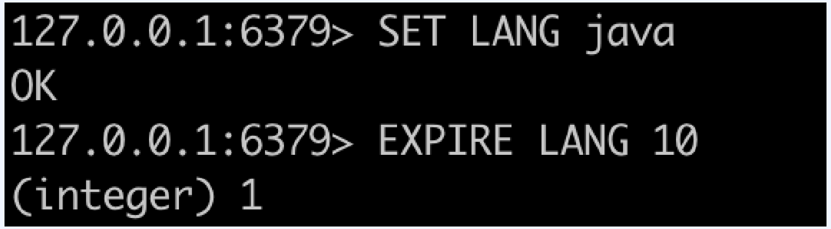


```
127.0.0.1:6379> SET LANG java
OK
127.0.0.1:6379> EXPIRE LANG 10
(integer) 1
```

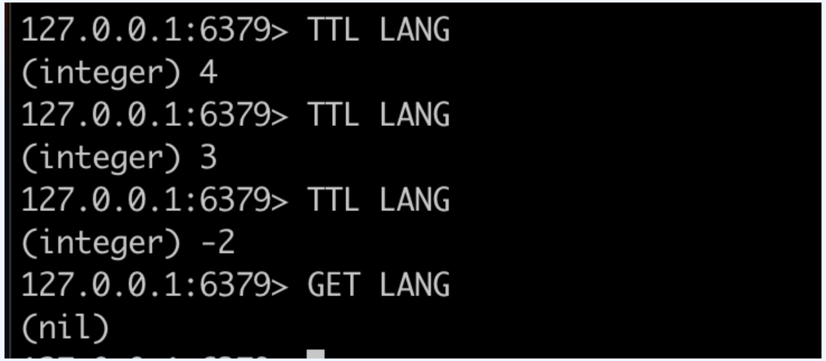
Data Type

Key 주요 명령어

TTL command

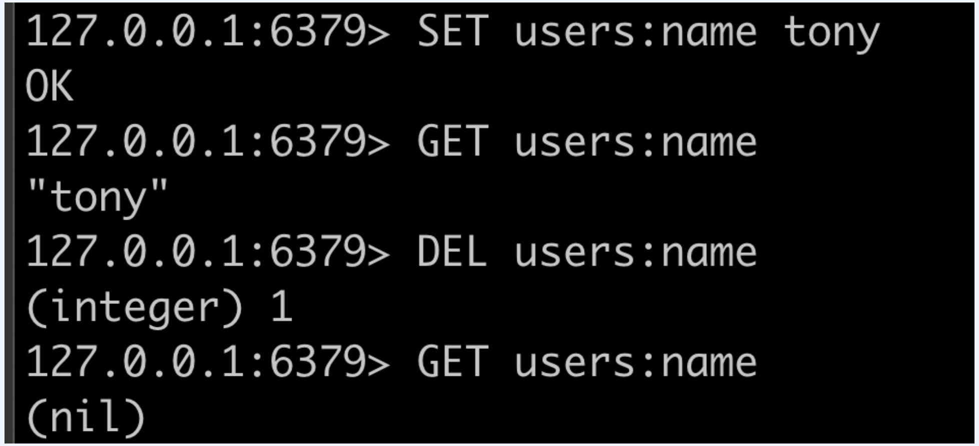


```
127.0.0.1:6379> SET LANG java
OK
127.0.0.1:6379> EXPIRE LANG 10
(integer) 1
```



```
127.0.0.1:6379> TTL LANG
(integer) 4
127.0.0.1:6379> TTL LANG
(integer) 3
127.0.0.1:6379> TTL LANG
(integer) -2
127.0.0.1:6379> GET LANG
(nil)
```

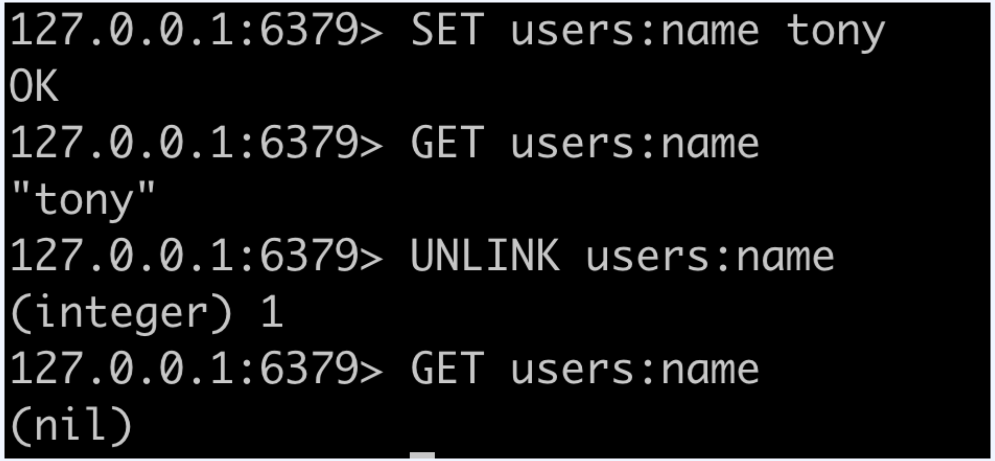
DEL command (sync)



```
127.0.0.1:6379> SET users:name tony
OK
127.0.0.1:6379> GET users:name
"tony"
127.0.0.1:6379> DEL users:name
(integer) 1
127.0.0.1:6379> GET users:name
(nil)
```

Key 주요 명령어

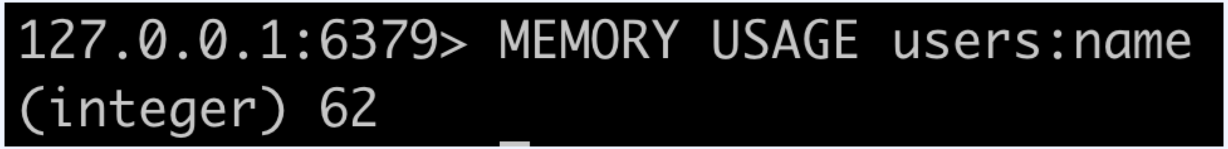
UNLINK command (async)



```
127.0.0.1:6379> SET users:name tony
OK
127.0.0.1:6379> GET users:name
"tony"
127.0.0.1:6379> UNLINK users:name
(integer) 1
127.0.0.1:6379> GET users:name
(nil)
```

Key 주요 명령어

MEMORY USAGE command



```
127.0.0.1:6379> MEMORY USAGE users:name
(integer) 62
```

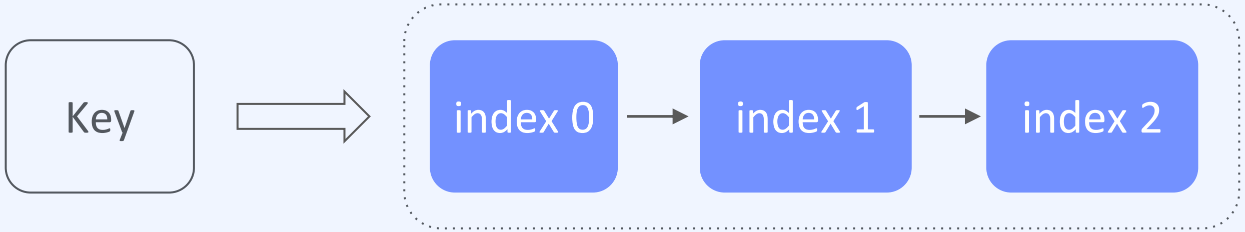
Data Type

Lists

Lists

Lists

- Linked List(strings)
    - 예) Java ArrayList
- Queue, Stack



Data Type

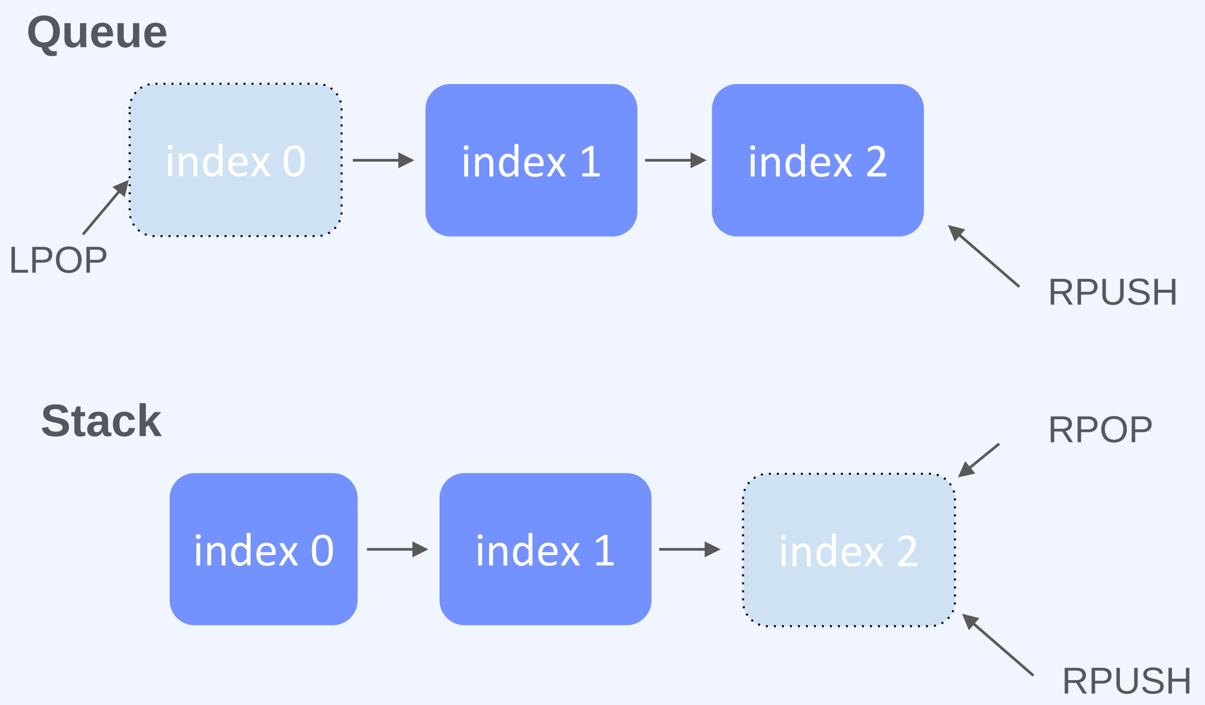
Lists

command

- LPUSH
- RPUSH
- LPOP
- RPOP
- LLEN
- LRANGE

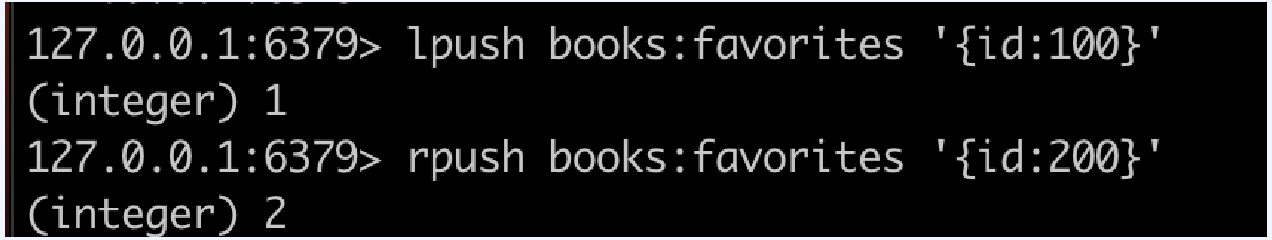
Data Type

Lists example

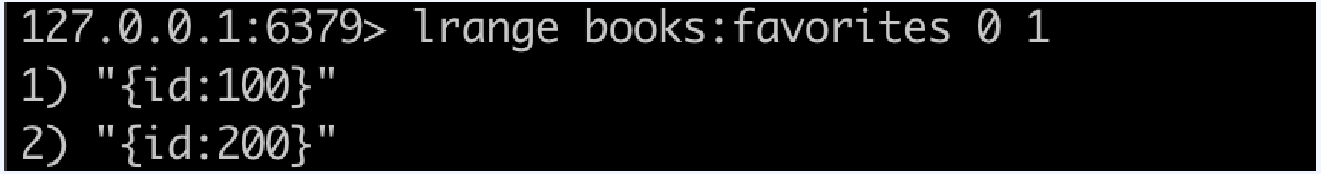


Data Type

Lists example



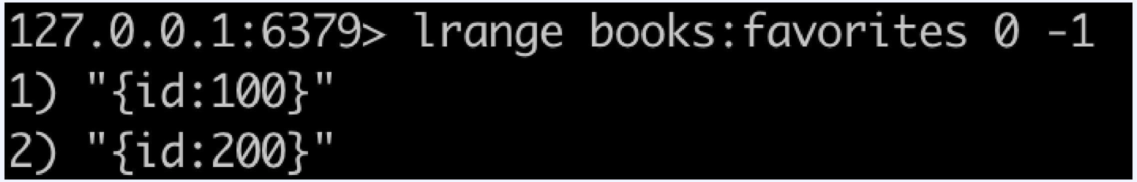
```
127.0.0.1:6379> lpush books:favorites '{id:100}'
(integer) 1
127.0.0.1:6379> rpush books:favorites '{id:200}'
(integer) 2
```



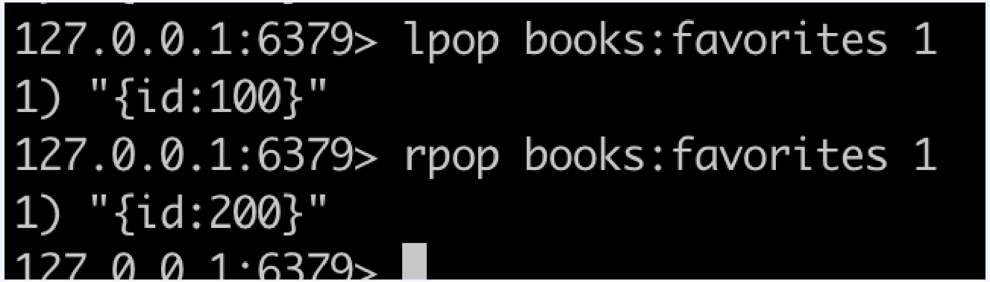
```
127.0.0.1:6379> lrange books:favorites 0 1
1) "{id:100}"
2) "{id:200}"
```

Data Type

Lists example



```
127.0.0.1:6379> lrange books:favorites 0 -1
1) "{id:100}"
2) "{id:200}"
```



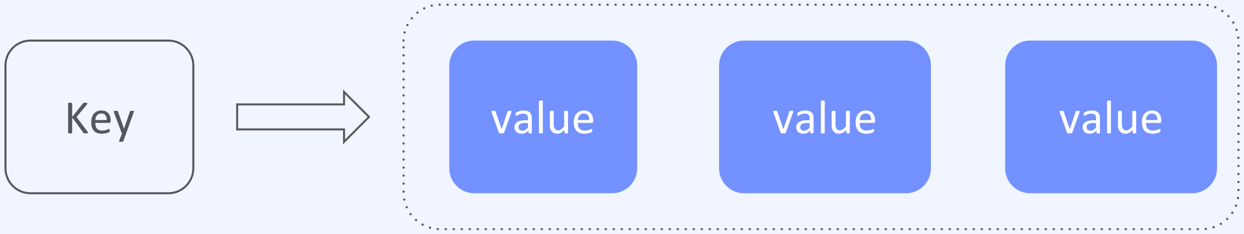
```
127.0.0.1:6379> lpop books:favorites 1
1) "{id:100}"
127.0.0.1:6379> rpop books:favorites 1
1) "{id:200}"
127.0.0.1:6379>
```

Data Type

Sets

Sets

- Unordered collection(unique strings)
  - 예) Java Set
- Unique item
  - SNS follow
  - Blacklist
  - Tags



Data Type

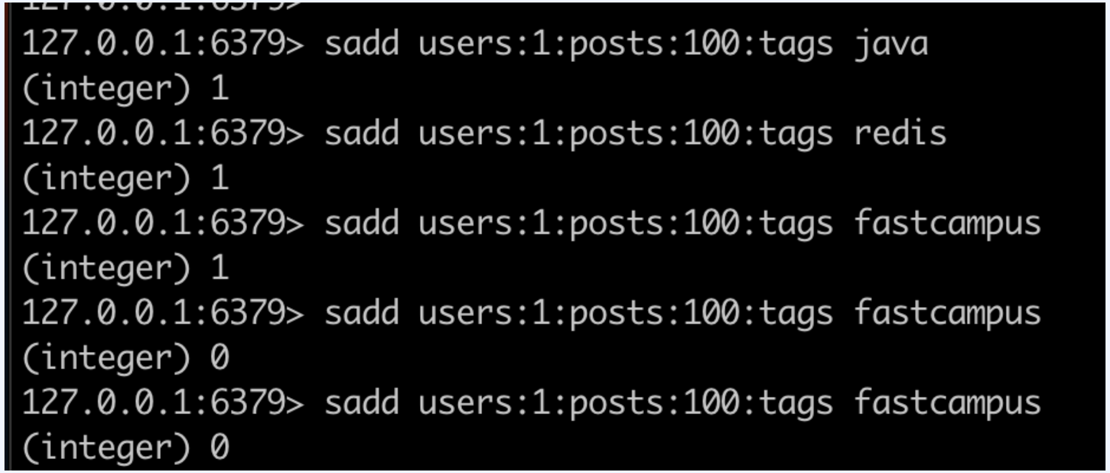
Sets

command

- SADD
- SREM
- SISMEMBER
- SMEMBERS
- SINTER
- SCARD

Data Type

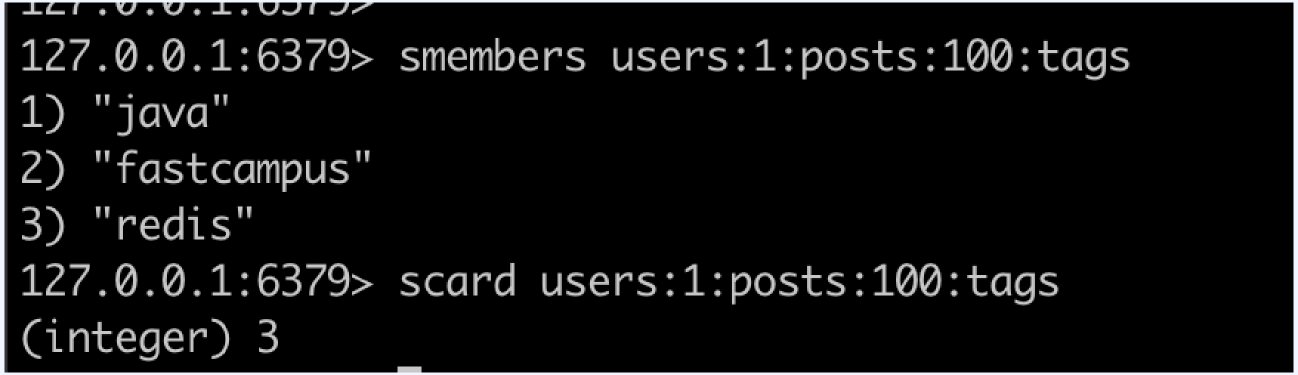
Sets example



```
127.0.0.1:6379> sadd users:1:posts:100:tags java
(integer) 1
127.0.0.1:6379> sadd users:1:posts:100:tags redis
(integer) 1
127.0.0.1:6379> sadd users:1:posts:100:tags fastcampus
(integer) 1
127.0.0.1:6379> sadd users:1:posts:100:tags fastcampus
(integer) 0
127.0.0.1:6379> sadd users:1:posts:100:tags fastcampus
(integer) 0
```

Data Type

Sets example



```
127.0.0.1:6379> smembers users:1:posts:100:tags
1) "java"
2) "fastcampus"
3) "redis"
127.0.0.1:6379> scard users:1:posts:100:tags
(integer) 3
```

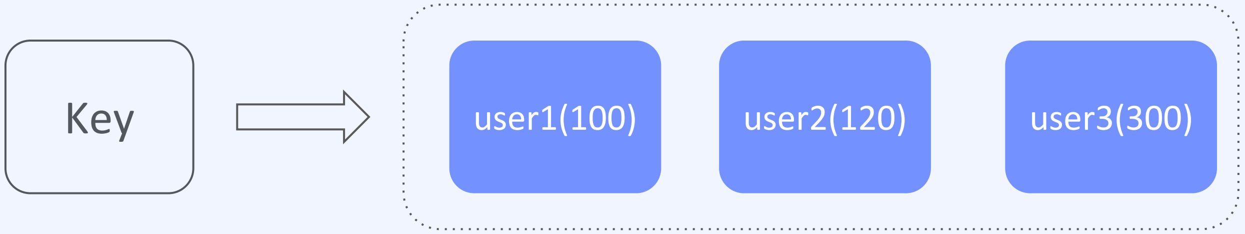
Data Type

Sorted Sets

Sorted Sets

Sorted Sets

- ordered collection(unique strings)
  - 예) Java SortedSet
- Leader board
- Rate limit



Data Type

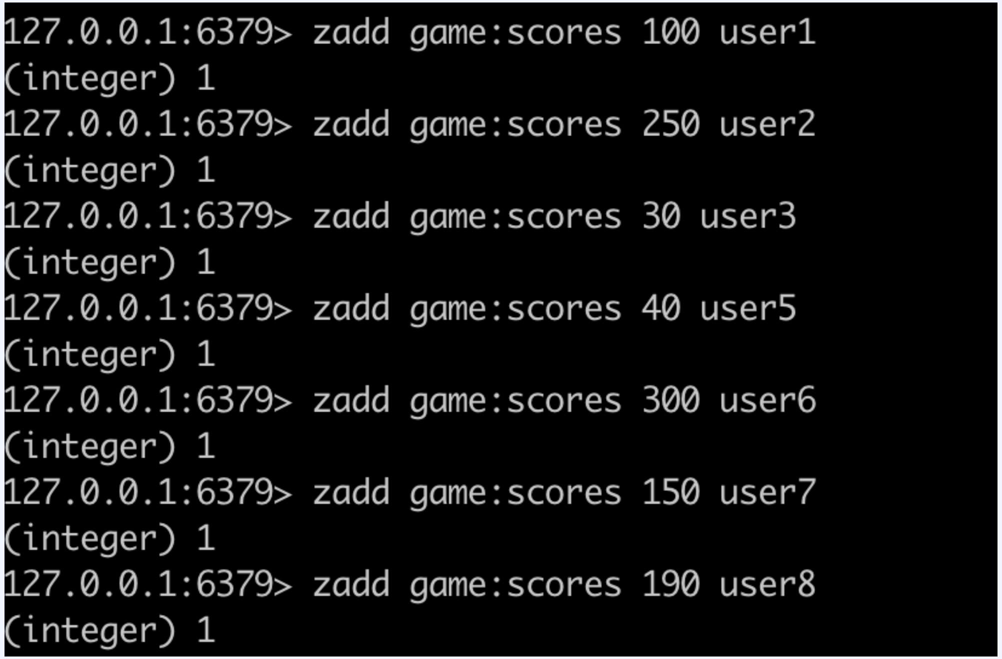
Sorted Sets

command

- ZADD
- ZREM
- ZRANGE (6.2: REV, BYSCORE, BYLEX and LIMIT option)
- ZCARD
- ZRANK / ZREVRANK
- ZINCRBY

Data Type

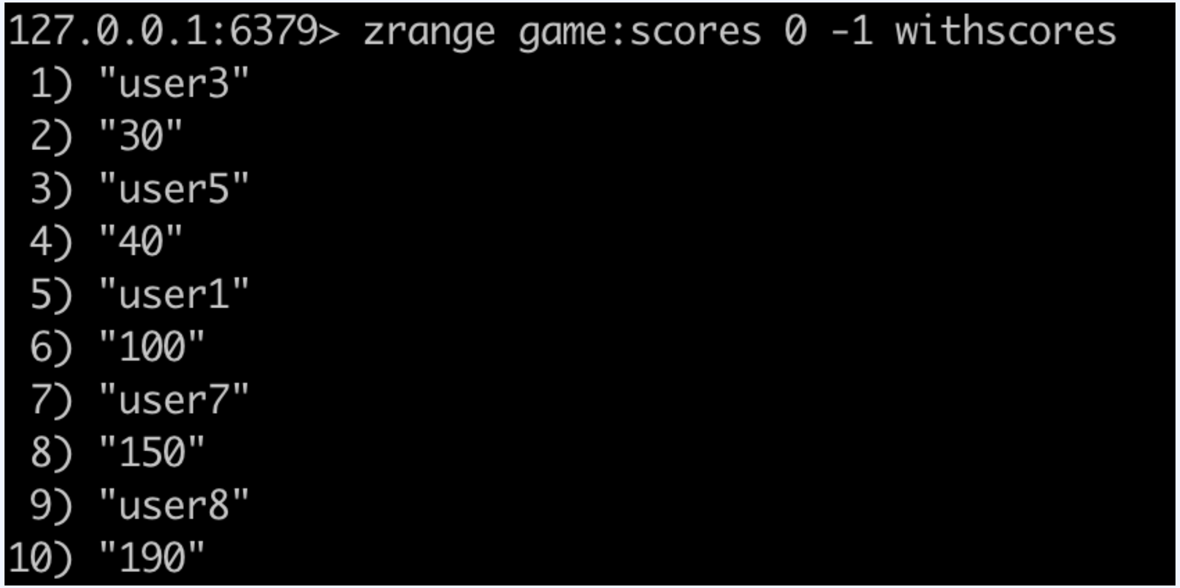
Sorted Sets example



```
127.0.0.1:6379> zadd game:scores 100 user1
(integer) 1
127.0.0.1:6379> zadd game:scores 250 user2
(integer) 1
127.0.0.1:6379> zadd game:scores 30 user3
(integer) 1
127.0.0.1:6379> zadd game:scores 40 user5
(integer) 1
127.0.0.1:6379> zadd game:scores 300 user6
(integer) 1
127.0.0.1:6379> zadd game:scores 150 user7
(integer) 1
127.0.0.1:6379> zadd game:scores 190 user8
(integer) 1
```

Data Type

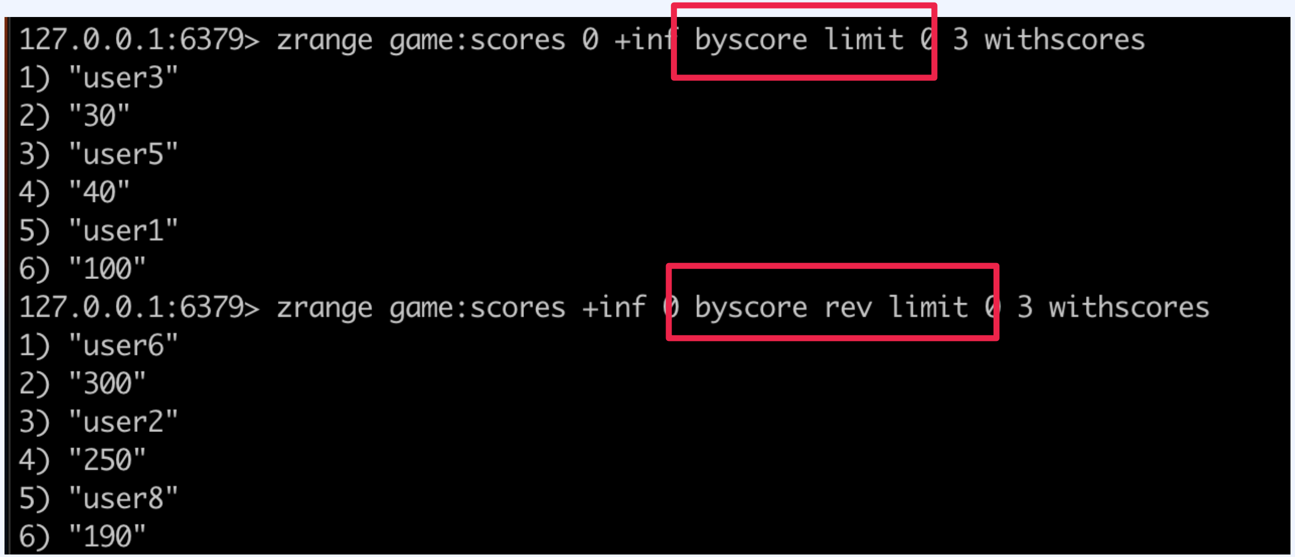
Sorted Sets example



```
127.0.0.1:6379> zrange game:scores 0 -1 withscores
 1) "user3"
 2) "30"
 3) "user5"
 4) "40"
 5) "user1"
 6) "100"
 7) "user7"
 8) "150"
 9) "user8"
10) "190"
```

Data Type

Sorted Sets example



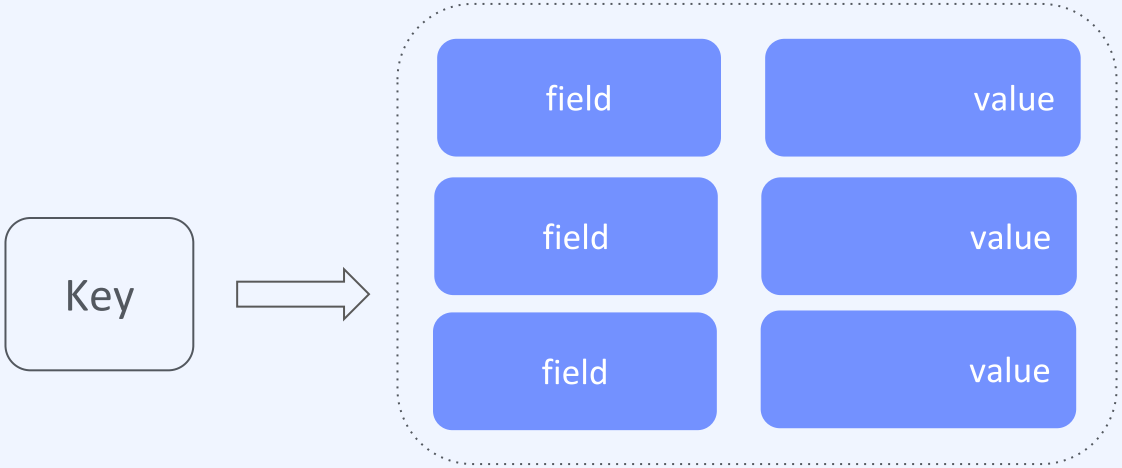
```
127.0.0.1:6379> zrange game:scores 0 +inf byscore limit 0 3 withscores
1) "user3"
2) "30"
3) "user5"
4) "40"
5) "user1"
6) "100"
127.0.0.1:6379> zrange game:scores +inf 0 byscore rev limit 0 3 withscores
1) "user6"
2) "300"
3) "user2"
4) "250"
5) "user8"
6) "190"
```

Data Type

Hashes

Hashes

- field-value pair collections
  - 예) Java HashMap



Data Type

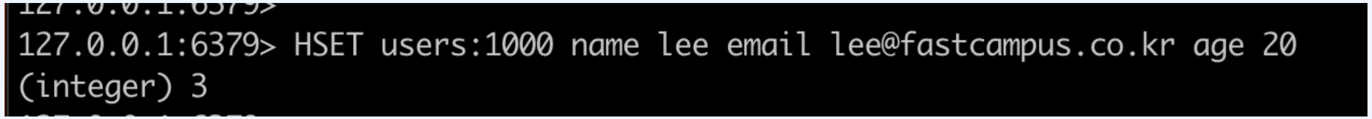
Hashes

command

- HSET
- HGET, HMGET
- HGETALL
- HDEL
- HINCRBY

Data Type

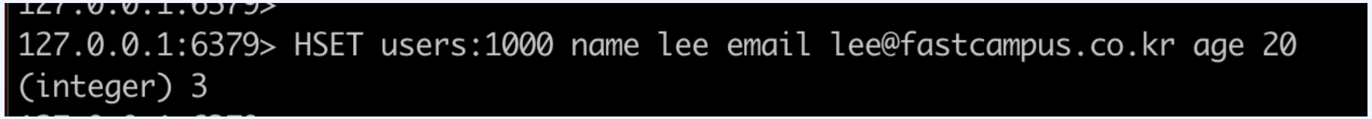
Hashes example



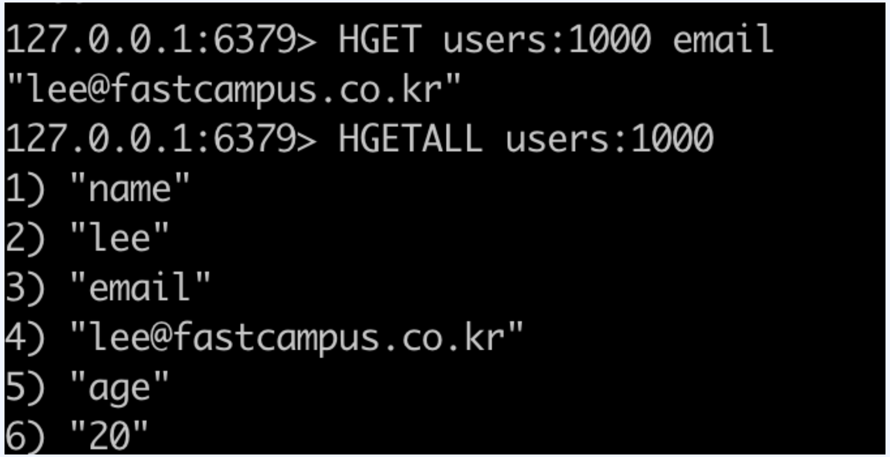
```
127.0.0.1:6379> HSET users:1000 name lee email lee@fastcampus.co.kr age 20
(integer) 3
```

Data Type

Hashes example



```
127.0.0.1:6379> HSET users:1000 name lee email lee@fastcampus.co.kr age 20
(integer) 3
```



```
127.0.0.1:6379> HGET users:1000 email
"lee@fastcampus.co.kr"
127.0.0.1:6379> HGETALL users:1000
1) "name"
2) "lee"
3) "email"
4) "lee@fastcampus.co.kr"
5) "age"
6) "20"
```

Data Type

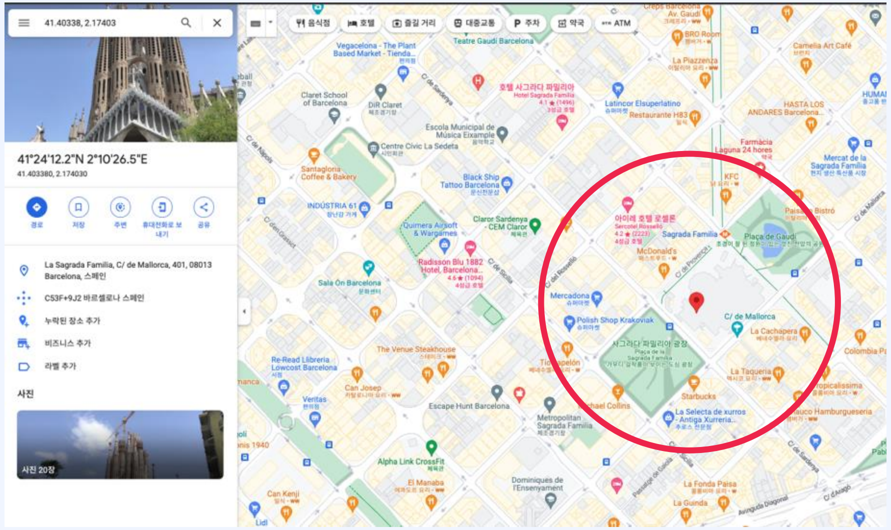
Geospatial

Data Type

Geospatial

Geospatial

- Coordinate (Latitude and Longitude)



출처:https://www.google.com/maps

Data Type

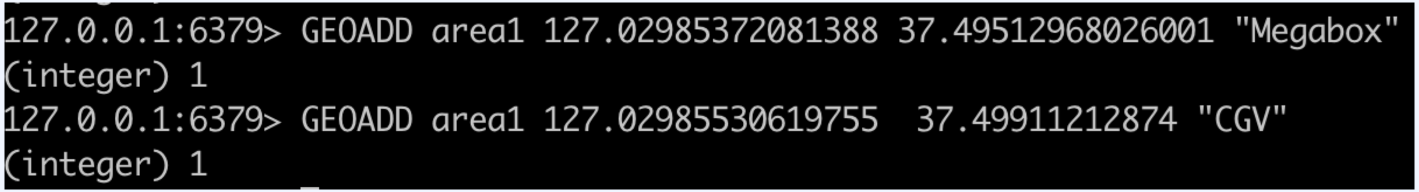
Geospatial

command

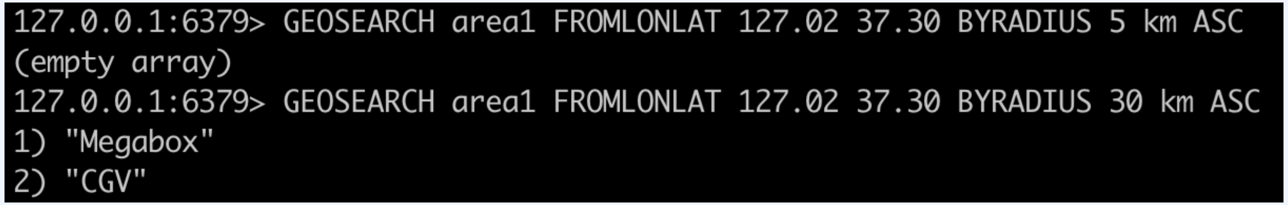
- GEOADD
- GEOSEARCH (6.2)
- GEODIST
- GEOPOS

Data Type

Geospatial example



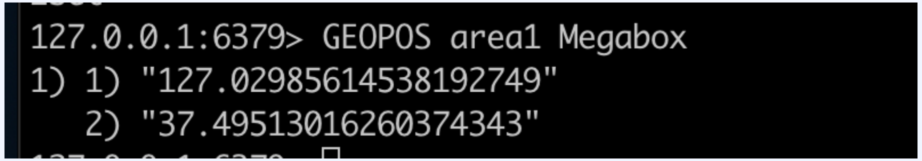
```
127.0.0.1:6379> GEOADD area1 127.02985372081388 37.49512968026001 "Megabox"
(integer) 1
127.0.0.1:6379> GEOADD area1 127.02985530619755  37.49911212874 "CGV"
(integer) 1
```



```
127.0.0.1:6379> GEOSEARCH area1 FROMLONLAT 127.02 37.30 BYRADIUS 5 km ASC
(empty array)
127.0.0.1:6379> GEOSEARCH area1 FROMLONLAT 127.02 37.30 BYRADIUS 30 km ASC
1) "Megabox"
2) "CGV"
```

Data Type

Geospatial example



```
127.0.0.1:6379> GEOPOS area1 Megabox
1) 1) "127.02985614538192749"
   2) "37.49513016260374343"
```

Data Type

Bitmap

Bitmap

Bitmap

- 0 또는 1의 값으로 이루어진 비트열
    - 메모리를 적게 사용하여 대량의 데이터 저장에 유용

Data Type

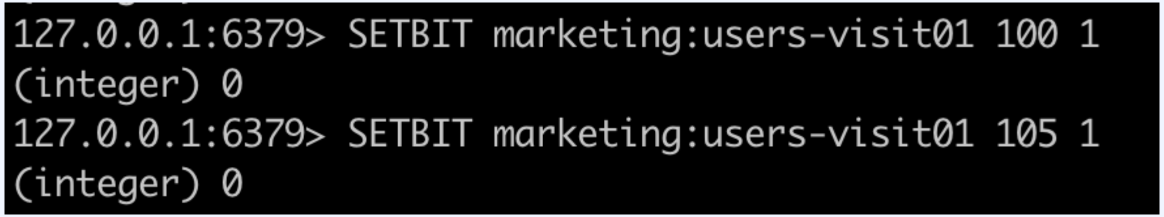
Bitmap

command

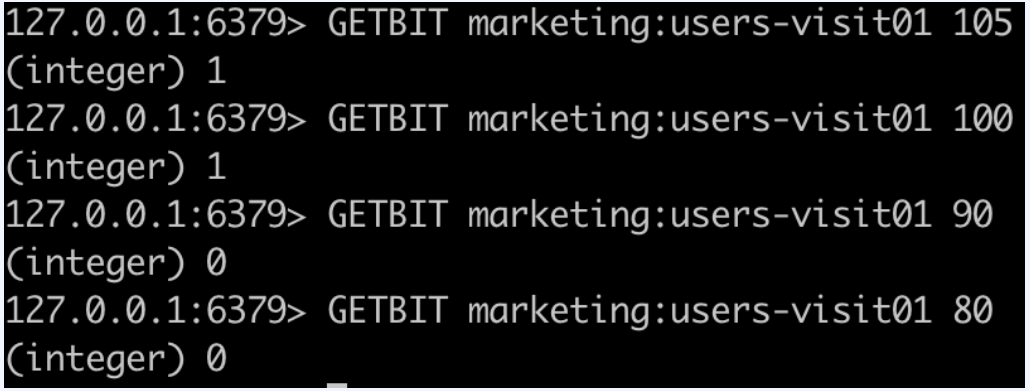
- SETBIT
- GETBIT
- BITCOUNT

Data Type

Bitmap example



```
127.0.0.1:6379> SETBIT marketing:users-visit01 100 1
(integer) 0
127.0.0.1:6379> SETBIT marketing:users-visit01 105 1
(integer) 0
```



```
127.0.0.1:6379> GETBIT marketing:users-visit01 105
(integer) 1
127.0.0.1:6379> GETBIT marketing:users-visit01 100
(integer) 1
127.0.0.1:6379> GETBIT marketing:users-visit01 90
(integer) 0
127.0.0.1:6379> GETBIT marketing:users-visit01 80
(integer) 0
```

마무리

Key/Value(Data type)

- Strings
- Lists
- Sets
- Sorted sets
- Hashes
- Geospatial
- Bitmap