Commands

Search commands...

Filter by group...

ACL CAT
Lists the ACL categories, or the commands inside a category.

ACL DELUSER
Deletes ACL users, and terminates their connections.

ACL DRYRUN
Simulates the execution of a command by a user, without executing the command.

ACL GENPASS
Generates a pseudorandom, secure password that can be used to identify ACL users.

ACL GETUSER
Lists the ACL rules of a user.

ACL LIST
Dumps the effective rules in ACL file format.

ACL LOAD
Reloads the rules from the configured ACL file.

ACL LOG
Lists recent security events generated due to ACL rules.

ACL SAVE
Saves the effective ACL rules in the configured ACL file.

ACL SETUSER
Creates and modifies an ACL user and its rules.

ACL USERS
Lists all ACL users.

ACL WHOAMI
Returns the authenticated username of the current connection.

APPEND
Appends a string to the value of a key. Creates the key if it doesn't exist.

ASKING
Signals that a cluster client is following an -ASK redirect.

AUTH
Authenticates the connection.

BF.ADD
Adds an item to a Bloom Filter

BF.CARD
Returns the cardinality of a Bloom filter

BF.EXISTS
Checks whether an item exists in a Bloom Filter

BF.INFO
Returns information about a Bloom Filter

BF.INSERT
Adds one or more items to a Bloom Filter. A filter will be created if it does not exist

BF.LOADCHUNK
Restores a filter previously saved using SCANDUMP

BF.MADD
Adds one or more items to a Bloom Filter. A filter will be created if it does not exist

BF.MEXISTS
Checks whether one or more items exist in a Bloom Filter

BF.RESERVE
Creates a new Bloom Filter

BF.SCANDUMP
Begins an incremental save of the bloom

BGREWRITEAOF
Asynchronously rewrites the append-

BGSAVE
Asynchronously saves the database(s)

A B C D E F G H I J K L M O P Q R S T U W X Z

filter

only file to disk.

to disk.

Counts the number of set bits (population counting) in a string.

Performs arbitrary bitfield integer operations on strings.

Performs arbitrary read-only bitfield integer operations on strings.

BITOP
Performs bitwise operations on multiple strings, and stores the result.

BITPOS
Finds the first set (1) or clear (0) bit in a string.

BLMOVE
Pops an element from a list, pushes it to another list and returns it. Blocks until an element is available otherwise. Deletes the list if the last element was moved.

BLMPOP
Pops the first element from one of multiple lists. Blocks until an element is available otherwise. Deletes the list if the last element was popped.

BLPOP
Removes and returns the first element in a list. Blocks until an element is available otherwise. Deletes the list if the last element was popped.

BRPOP
Removes and returns the last element in a list. Blocks until an element is available otherwise. Deletes the list if the last element was popped.

BRPOPLPUSH
Pops an element from a list, pushes it to another list and returns it. Block until an element is available otherwise. Deletes the list if the last element was popped.

BZMPOP
Removes and returns a member by score from one or more sorted sets. Blocks until a member is available otherwise. Deletes the sorted set if the last element was popped.

BZPOPMAX
Removes and returns the member with the highest score from one or more sorted sets. Blocks until a member available otherwise. Deletes the sorted set if the last element was popped.

BZPOPMIN
Removes and returns the member with the lowest score from one or more sorted sets. Blocks until a member is available otherwise. Deletes the sorted set if the last element was popped.

CF.ADD
Adds an item to a Cuckoo Filter

CF.ADDNX
Adds an item to a Cuckoo Filter if the item did not exist previously.

CF.COUNT
Return the number of times an item might be in a Cuckoo Filter

CF.DEL
Deletes an item from a Cuckoo Filter

CF.EXISTS
Checks whether one or more items exist in a Cuckoo Filter

CF.INFO
Returns information about a Cuckoo Filter

CF.INSERT
Adds one or more items to a Cuckoo Filter. A filter will be created if it does not exist

CF.INSERTNX
Adds one or more items to a Cuckoo Filter if the items did not exist previously. A filter will be created if it does not exist

CF.LOADCHUNK
Restores a filter previously saved using SCANDUMP

CF.MEXISTS
Checks whether one or more items exist in a Cuckoo Filter

CF.RESERVE
Creates a new Cuckoo Filter

CF.SCANDUMP
Begins an incremental save of the bloom filter

CLIENT CACHING
Instructs the server whether to track the keys in the next request.

CLIENT GETNAME
Returns the name of the connection.

CLIENT GETREDIR
Returns the client ID to which the connection's tracking notifications are redirected.

CLIENT ID
Returns the unique client ID of the connection.

CLIENT INFO
Returns information about the connection.

CLIENT KILL
Terminates open connections.

CLIENT LIST
Lists open connections.

CLIENT NO-EVICT
Sets the client eviction mode of the connection.

CLIENT NO-TOUCH
Controls whether commands sent by the client affect the LRU/LFU of accessed keys.

CLIENT PAUSE
Suspends commands processing.

CLIENT REPLY
Instructs the server whether to reply to commands.

CLIENT SETINFO
Sets information specific to the client or connection.

CLIENT SETNAME
Sets the connection name.

CLIENT TRACKING
Controls server-assisted client-side caching for the connection.

CLIENT TRACKINGINFO
Returns information about server-assisted client-side caching for the connection.

CLIENT UNBLOCK
Unblocks a client blocked by a blocking command from a different connection.

CLIENT UNPAUSE
Resumes processing commands from paused clients.

CLUSTER ADDSLOTS
Assigns new hash slots to a node.

CLUSTER ADDSLOTSRANGE
Assigns new hash slot ranges to a node.

CLUSTER BUMPEPOCH
Advances the cluster config epoch.

CLUSTER COUNT-FAILURE-REPORTS
Returns the number of active failure reports active for a node.

CLUSTER COUNTKEYSINSLOT
Returns the number of keys in a hash slot.

CLUSTER DELSLOTS
Sets hash slots as unbound for a node.

CLUSTER DELSLOTSRANGE
Sets hash slot ranges as unbound for a node.

CLUSTER FAILOVER
Forces a replica to perform a manual failover of its master.

CLUSTER FLUSHSLOTS
Deletes all slots information from a node.

CLUSTER FORGET
Removes a node from the nodes table.

CLUSTER GETKEYSINSLOT
Returns the key names in a hash slot.

CLUSTER INFO
Returns information about the state of a node.

CLUSTER KEYSLOT
Returns the hash slot for a key.

CLUSTER LINKS
Returns a list of all TCP links to and from peer nodes.

CLUSTER MEET
Forces a node to handshake with another node.

CLUSTER MYID
Returns the ID of a node.

CLUSTER MYSHARDID
Returns the shard ID of a node.

CLUSTER NODES
Returns the cluster configuration for a node.

CLUSTER REPLICAS
Lists the replica nodes of a master node.

CLUSTER REPLICATE
Configure a node as replica of a master node.

CLUSTER RESET
Resets a node.

CLUSTER SAVECONFIG
Forces a node to save the cluster configuration to disk.

CLUSTER SET-CONFIG-EPOCH
Sets the configuration epoch for a new node.

CLUSTER SETSLOT
Binds a hash slot to a node.

CLUSTER SHARDS
Returns the mapping of cluster slots to shards.

CLUSTER SLAVES
Lists the replica nodes of a master node.

CLUSTER SLOTS
Returns the mapping of cluster slots to nodes.

CMS.INCRBY
Increases the count of one or more items by increment

CMS.INFO
Returns information about a sketch

CMS.INITBYDIM
Initializes a Count-Min Sketch to dimensions specified by user

CMS.INITBYPROB
Initializes a Count-Min Sketch to accommodate requested tolerances.

CMS.MERGE
Merges several sketches into one sketch

CMS.QUERY
Returns the count for one or more items in a sketch

COMMAND
Returns detailed information about all commands.

COMMAND COUNT
Returns a count of commands.

COMMAND DOCS
Returns documentary information about one, multiple or all commands.

COMMAND GETKEYS
Extracts the key names from an arbitrary command.

COMMAND GETKEYSANDFLAGS
Extracts the key names and access flags for an arbitrary command.

COMMAND INFO
Returns information about one, multiple or all commands.

COMMAND LIST
Returns a list of command names.

CONFIG GET
Returns the effective values of configuration parameters.

CONFIG RESETSTAT
Resets the server's statistics.

CONFIG REWRITE
Persists the effective configuration to file.

CONFIG SET
Sets configuration parameters in-flight.

COPY
Copies the value of a key to a new key.

DBSIZE
Returns the number of keys in the database.

DECR
Decrements the integer value of a key by one. Uses 0 as initial value if the key doesn't exist.

DECRBY
Decrements a number from the integer value of a key. Uses 0 as initial value if the key doesn't exist.

DEL
Deletes one or more keys.

DISCARD
Discards a transaction.

DUMP
Returns a serialized representation of

the value stored at a key.

ECHO
Returns the given string.

EVAL
Executes a server-side Lua script.

EVAL_RO
Executes a read-only server-side Lua script.

EVALSHA
Executes a server-side Lua script by SHA1 digest.

EVALSHA_RO
Executes a read-only server-side Lua script by SHA1 digest.

EXEC
Executes all commands in a transaction.

EXISTS
Determines whether one or more keys exist.

EXPIRE
Sets the expiration time of a key in seconds.

EXPIREAT
Sets the expiration time of a key to a Unix timestamp.

EXPIRETIME
Returns the expiration time of a key as a Unix timestamp.

FAILOVER
Starts a coordinated failover from a server to one of its replicas.

FCALL
Invokes a function.

FCALL_RO
Invokes a read-only function.

FLUSHALL
Removes all keys from all databases.

FLUSHDB
Remove all keys from the current database.

FT._LIST
Returns a list of all existing indexes

FT.AGGREGATE
Run a search query on an index and perform aggregate transformations on the results

FT.ALIASADD
Adds an alias to the index

FT.ALIASDEL
Deletes an alias from the index

FT.ALIASUPDATE
Adds or updates an alias to the index

FT.ALTER
Adds a new field to the index

FT.CONFIG GET
Retrieves runtime configuration options

FT.CONFIG SET
Sets runtime configuration options

FT.CREATE
Creates an index with the given spec

FT.CURSOR DEL
Deletes a cursor

FT.CURSOR READ
Reads from a cursor

FT.DICTADD
Adds terms to a dictionary

FT.DICTDEL
Deletes terms from a dictionary

FT.DICTDUMP
Dumps all terms in the given dictionary

FT.DROPINDEX
Deletes the index

FT.EXPLAIN
Returns the execution plan for a complex query

FT.EXPLAINCLI
Returns the execution plan for a complex query

FT.INFO
Returns information and statistics on the index

FT.PROFILE
Performs a `FT.SEARCH` or `FT.AGGREGATE` command and collects performance information

FT.SEARCH
Searches the index with a textual query, returning either documents or just ids

FT.SPELLCHECK
Performs spelling correction on a query, returning suggestions for misspelled terms

FT.SUGADD
Adds a suggestion string to an auto-complete suggestion dictionary

FT.SUGDEL
Deletes a string from a suggestion index

FT.SUGGET
Gets completion suggestions for a prefix

FT.SUGLEN
Gets the size of an auto-complete suggestion dictionary

FT.SYNDUMP
Dumps the contents of a synonym group

FT.SYNUPDATE
Creates or updates a synonym group with additional terms

FT.TAGVALS
Returns the distinct tags indexed in a Tag field

FUNCTION DELETE
Deletes a library and its functions.

FUNCTION DUMP
Dumps all libraries into a serialized binary payload.

FUNCTION FLUSH
Deletes all libraries and functions.

FUNCTION KILL
Terminates a function during execution.

FUNCTION LIST
Returns information about all libraries.

FUNCTION LOAD
Creates a library.

FUNCTION RESTORE
Restores all libraries from a payload.

FUNCTION STATS
Returns information about a function during execution.

GEOADD
Adds one or more members to a geospatial index. The key is created if it doesn't exist.

GEODIST
Returns the distance between two members of a geospatial index.

GEOHASH
Returns members from a geospatial index as geohash strings.

GEOPOS
Returns the longitude and latitude of members from a geospatial index.

GEORADIUS
Queries a geospatial index for members within a distance from a coordinate, optionally stores the result.

GEORADIUS_RO
Returns members from a geospatial index that are within a distance from a coordinate.

GEORADIUSBYMEMBER
Queries a geospatial index for members within a distance from a member, optionally stores the result.

GEORADIUSBYMEMBER_RO
Returns members from a geospatial index that are within a distance from a member.

GEOSEARCH
Queries a geospatial index for members inside an area of a box or a circle.

GEOSEARCHSTORE
Queries a geospatial index for members inside an area of a box or a circle, optionally stores the result.

GET
Returns the string value of a key.

GETBIT
Returns a bit value by offset.

GETDEL
Returns the string value of a key after deleting the key.

GETEX
Returns the string value of a key after setting its expiration time.

GETRANGE
Returns a substring of the string stored at a key.

GETSET
Returns the previous string value of a key after setting it to a new value.

HDEL
Deletes one or more fields and their values from a hash. Deletes the hash if no fields remain.

HELLO
Handshakes with the Redis server.

HEXISTS
Determines whether a field exists in a hash.

HGET
Returns the value of a field in a hash.

HGETALL
Returns all fields and values in a hash.

HINCRBY
Increments the integer value of a field in a hash by a number. Uses 0 as initial value if the field doesn't exist.

HINCRBYFLOAT
Increments the floating point value of a field by a number. Uses 0 as initial value if the field doesn't exist.

HKEYS
Returns all fields in a hash.

HLEN
Returns the number of fields in a hash.

HMGET
Returns the values of all fields in a hash.

HMSET
Sets the values of multiple fields.

HRANDFIELD
Returns one or more random fields from a hash.

HSCAN
Iterates over fields and values of a hash.

HSET
Creates or modifies the value of a field in a hash.

HSETNX
Sets the value of a field in a hash only when the field doesn't exist.

HSTRLEN
Returns the length of the value of a field.

HVALS
Returns all values in a hash.

INCR
Increments the integer value of a key by one. Uses 0 as initial value if the key doesn't exist.

INCRBY
Increments the integer value of a key by a number. Uses 0 as initial value if the key doesn't exist.

INCRBYFLOAT
Increment the floating point value of a key by a number. Uses 0 as initial value if the key doesn't exist.

INFO
Returns information and statistics about the server.

JSON.ARRAPPEND
Append one or more json values into the array at path after the last element in it.

JSON.ARRINDEX
Returns the index of the first occurrence of a JSON scalar value in the array at path

JSON.ARRINSERT
Inserts the JSON scalar(s) value at the specified index in the array at path

JSON.ARRLEN
Returns the length of the array at path

JSON.ARRPOP
Removes and returns the element at the specified index in the array at path

JSON.ARRTRIM
Trims the array at path to contain only the specified inclusive range of indices from start to stop

JSON.CLEAR
Clears all values from an array or an object and sets numeric values to `0`

JSON.DEBUG
Debugging container command

JSON.DEBUG MEMORY
Reports the size in bytes of a key

JSON.DEL
Deletes a value

JSON.FORGET
Deletes a value

JSON.GET
Gets the value at one or more paths in JSON serialized form

JSON.MERGE
Merges a given JSON value into matching paths. Consequently, JSON values at matching paths are updated, deleted, or expanded with new children

JSON.MGET
Returns the values at a path from one or more keys

JSON.MSET
Sets or updates the JSON value of one or more keys

JSON.NUMINCRBY
Increments the numeric value at path by a value

JSON.NUMMULTBY
Multiplies the numeric value at path by a value

JSON.OBJKEYS
Returns the JSON keys of the object at path

JSON.OBJLEN
Returns the number of keys of the object at path

JSON.RESP
Returns the JSON value at path in Redis Serialization Protocol (RESP)

JSON.SET
Sets or updates the JSON value at a path

JSON.STRAPPEND
Appends a string to a JSON string value at path

JSON.STRLEN
Returns the length of the JSON String at path in key

JSON.TOGGLE
Toggles a boolean value

JSON.TYPE
Returns the type of the JSON value at path

KEYS
Returns all key names that match a pattern.

LASTSAVE
Returns the Unix timestamp of the last successful save to disk.

LATENCY DOCTOR
Returns a human-readable latency analysis report.

LATENCY GRAPH
Returns a latency graph for an event.

LATENCY HISTOGRAM
Returns the cumulative distribution of latencies of a subset or all commands.

LATENCY HISTORY
Returns timestamp-latency samples for an event.

LATENCY LATEST
Returns the latest latency samples for all events.

LATENCY RESET
Resets the latency data for one or more events.

LCS
Finds the longest common substring.

LINDEX
Returns an element from a list by its index.

LINSERT
Inserts an element before or after another element in a list.

LLEN
Returns the length of a list.

LMOVE
Returns an element after popping it from one list and pushing it to another. Deletes the list if the last element was moved.

LMPOP
Returns multiple elements from a list after removing them. Deletes the list if the last element was popped.

LOLWUT
Displays computer art and the Redis version

LPOP
Returns the first elements in a list after removing it. Deletes the list if the last element was popped.

LPOS
Returns the index of matching elements in a list.

LPUSH
Prepends one or more elements to a list. Creates the key if it doesn't exist.

LPUSHX
Prepends one or more elements to a list only when the list exists.

LRANGE
Returns a range of elements from a list.

LREM
Removes elements from a list. Deletes the list if the last element was removed.

LSET
Sets the value of an element in a list by its index.

LTRIM
Removes elements from both ends a list. Deletes the list if all elements were trimmed.

MEMORY DOCTOR
Outputs a memory problems report.

MEMORY MALLOC-STATS
Returns the allocator statistics.

MEMORY PURGE
Asks the allocator to release memory.

MEMORY STATS
Returns details about memory usage.

MEMORY USAGE
Estimates the memory usage of a key.

MGET
Atomically returns the string values of one or more keys.

MIGRATE
Atomically transfers a key from one Redis instance to another.

MODULE LIST
Returns all loaded modules.

MODULE LOAD
Loads a module.

MODULE LOADEX
Loads a module using extended parameters.

MODULE UNLOAD
Unloads a module.

MONITOR
Listens for all requests received by the server in real-time.

MOVE
Moves a key to another database.

MSET
Atomically creates or modifies the string values of one or more keys.

MSETNX
Atomically modifies the string values of one or more keys only when all keys don't exist.

MULTI
Starts a transaction.

OBJECT ENCODING
Returns the internal encoding of a Redis object.

OBJECT FREQ
Returns the logarithmic access frequency counter of a Redis object.

OBJECT IDLETIME
Returns the time since the last access to a Redis object.

OBJECT REFCOUNT
Returns the reference count of a value of a key.

PERSIST
Removes the expiration time of a key.

PEXPIRE
Sets the expiration time of a key in milliseconds.

PEXPIREAT
Sets the expiration time of a key to a Unix milliseconds timestamp.

PEXPIRETIME
Returns the expiration time of a key as a

PFADD
Adds elements to a HyperLogLog key.

PFCOUNT
Returns the approximated cardinality of

Unix milliseconds timestamp.

Creates the key if it doesn't exist.

the set(s) observed by the HyperLogLog key(s).

PFDEBUG
Internal commands for debugging HyperLogLog values.

PFMERGE
Merges one or more HyperLogLog values into a single key.

PFSELFTEST
An internal command for testing HyperLogLog values.

PING
Returns the server's liveliness response.

PSETEX
Sets both string value and expiration time in milliseconds of a key. The key is created if it doesn't exist.

PSUBSCRIBE
Listens for messages published to channels that match one or more patterns.

PSYNC
An internal command used in replication.

PTTL
Returns the expiration time in milliseconds of a key.

PUBLISH
Posts a message to a channel.

PUBSUB CHANNELS
Returns the active channels.

PUBSUB NUMPAT
Returns a count of unique pattern subscriptions.

PUBSUB NUMSUB
Returns a count of subscribers to channels.

PUBSUB SHARDCHANNELS
Returns the active shard channels.

PUBSUB SHARDNUMSUB
Returns the count of subscribers of shard channels.

PUNSUBSCRIBE
Stops listening to messages published to channels that match one or more patterns.

QUIT
Closes the connection.

RANDOMKEY
Returns a random key name from the database.

READONLY
Enables read-only queries for a connection to a Redis Cluster replica node.

READWRITE
Enables read-write queries for a connection to a Reids Cluster replica node.

RENAME
Renames a key and overwrites the destination.

RENAMENX
Renames a key only when the target key name doesn't exist.

REPLCONF
An internal command for configuring the replication stream.

REPLICAOF
Configures a server as replica of another, or promotes it to a master.

RESET
Resets the connection.

RESTORE
Creates a key from the serialized representation of a value.

RESTORE-ASKING
An internal command for migrating keys in a cluster.

ROLE
Returns the replication role.

RPOP
Returns and removes the last elements of a list. Deletes the list if the last element was popped.

RPOPLPUSH
Returns the last element of a list after removing and pushing it to another list. Deletes the list if the last element was popped.

RPUSH
Appends one or more elements to a list. Creates the key if it doesn't exist.

RPUSHX
Appends an element to a list only when the list exists.

SADD
Adds one or more members to a set. Creates the key if it doesn't exist.

SAVE
Synchronously saves the database(s) to disk.

SCAN
Iterates over the key names in the database.

SCARD
Returns the number of members in a set.

SCRIPT DEBUG
Sets the debug mode of server-side Lua scripts.

SCRIPT EXISTS
Determines whether server-side Lua scripts exist in the script cache.

SCRIPT FLUSH
Removes all server-side Lua scripts from the script cache.

SCRIPT KILL
Terminates a server-side Lua script during execution.

SCRIPT LOAD
Loads a server-side Lua script to the script cache.

SDIFF
Returns the difference of multiple sets.

SDIFFSTORE
Stores the difference of multiple sets in a key.

SELECT
Changes the selected database.

SET
Sets the string value of a key, ignoring its type. The key is created if it doesn't exist.

SETBIT
Sets or clears the bit at offset of the string value. Creates the key if it doesn't exist.

SETEX
Sets the string value and expiration time of a key. Creates the key if it doesn't exist.

SETNX
Set the string value of a key only when the key doesn't exist.

SETRANGE
Overwrites a part of a string value with another by an offset. Creates the key if it doesn't exist.

SHUTDOWN
Synchronously saves the database(s) to disk and shuts down the Redis server.

SINTER
Returns the intersect of multiple sets.

SINTERCARD
Returns the number of members of the intersect of multiple sets.

SINTERSTORE
Stores the intersect of multiple sets in a key.

SISMEMBER
Determines whether a member belongs to a set.

SLAVEOF
Sets a Redis server as a replica of another, or promotes it to being a master.

SLOWLOG GET
Returns the slow log's entries.

SLOWLOG LEN
Returns the number of entries in the slow log.

SLOWLOG RESET
Clears all entries from the slow log.

SMEMBERS
Returns all members of a set.

SMISMEMBER
Determines whether multiple members belong to a set.

SMOVE
Moves a member from one set to another.

SORT
Sorts the elements in a list, a set, or a sorted set, optionally storing the result.

SORT_RO
Returns the sorted elements of a list, a set, or a sorted set.

SPOP
Returns one or more random members from a set after removing them. Deletes the set if the last member was popped.

SPUBLISH
Post a message to a shard channel

SRANDMEMBER
Get one or multiple random members from a set

SREM
Removes one or more members from a set. Deletes the set if the last member was removed.

SSCAN
Iterates over members of a set.

SSUBSCRIBE
Listens for messages published to shard channels.

STRLEN
Returns the length of a string value.

SUBSCRIBE
Listens for messages published to channels.

SUBSTR
Returns a substring from a string value.

SUNION
Returns the union of multiple sets.

SUNIONSTORE
Stores the union of multiple sets in a key.

SUNSUBSCRIBE
Stops listening to messages posted to shard channels.

SWAPDB
Swaps two Redis databases.

SYNC
An internal command used in replication.

TDIGEST.ADD
Adds one or more observations to a t-digest sketch

TDIGEST.BYRANK
Returns, for each input rank, an estimation of the value (floating-point) with that rank

TDIGEST.BYREVRANK
Returns, for each input reverse rank, an estimation of the value (floating-point) with that reverse rank

TDIGEST.CDF
Returns, for each input value, an estimation of the fraction (floating-point) of (observations smaller than the given value + half the observations equal to the given value)

TDIGEST.CREATE
Allocates memory and initializes a new t-digest sketch

TDIGEST.INFO
Returns information and statistics about a t-digest sketch

TDIGEST.MAX
Returns the maximum observation value from a t-digest sketch

TDIGEST.MERGE
Merges multiple t-digest sketches into a single sketch

TDIGEST.MIN
Returns the minimum observation value from a t-digest sketch

TDIGEST.QUANTILE
Returns, for each input fraction, an estimation of the value (floating point) that is smaller than the given fraction of observations

TDIGEST.RANK
Returns, for each input value (floating-point), the estimated rank of the value (the number of observations in the sketch that are smaller than the value + half the number of observations that are equal to the value)

TDIGEST.RESET
Resets a t-digest sketch: empty the sketch and re-initializes it.

TDIGEST.REVRANK
Returns, for each input value (floating-point), the estimated reverse rank of the

TDIGEST.TRIMMED_MEAN
Returns an estimation of the mean value from the sketch, excluding observation

value (the number of observations in the sketch that are larger than the value + half the number of observations that are equal to the value)

values outside the low and high cutoff quantiles

TFCALL
Invoke a JavaScript function

TFCALLASYNC
Invoke an asynchronous JavaScript function

TFUNCTION DELETE
Delete a JavaScript library from Redis by name

TFUNCTION LIST
List all JavaScript libraries loaded into Redis

TFUNCTION LOAD
Load a new JavaScript library into Redis

TIME
Returns the server time.

TOPK.ADD
Increases the count of one or more items by increment

TOPK.COUNT
Return the count for one or more items are in a sketch

TOPK.INCRBY
Increases the count of one or more items by increment

TOPK.INFO
Returns information about a sketch

TOPK.LIST
Return full list of items in Top K list

TOPK.QUERY
Checks whether one or more items are in a sketch

TOPK.RESERVE
Initializes a TopK with specified parameters

TOUCH
Returns the number of existing keys out of those specified after updating the time they were last accessed.

TS.ADD
Append a sample to a time series

TS.ALTER
Update the retention, chunk size, duplicate policy, and labels of an existing time series

TS.CREATE
Create a new time series

TS.CREATERULE
Create a compaction rule

TS.DECRBY
Decrease the value of the sample with the maximum existing timestamp, or create a new sample with a value equal to the value of the sample with the maximum existing timestamp with a given decrement

TS.DEL
Delete all samples between two timestamps for a given time series

TS.DELETERULE
Delete a compaction rule

TS.GET
Get the sample with the highest timestamp from a given time series

TS.INCRBY
Increase the value of the sample with the maximum existing timestamp, or create a new sample with a value equal to the value of the sample with the maximum existing timestamp with a given increment

TS.INFO
Returns information and statistics for a time series

TS.MADD
Append new samples to one or more time series

TS.MGET
Get the sample with the highest timestamp from each time series

TS.MRANGE
Query a range across multiple time series by filters in forward direction

matching a specific filter

TS.MREVRANGE
Query a range across multiple time-series by filters in reverse direction

TS.QUERYINDEX
Get all time series keys matching a filter list

TS.RANGE
Query a range in forward direction

TS.REVRANGE
Query a range in reverse direction

TTL
Returns the expiration time in seconds of a key.

TYPE
Determines the type of value stored at a key.

UNLINK
Asynchronously deletes one or more keys.

UNSUBSCRIBE
Stops listening to messages posted to channels.

UNWATCH
Forgets about watched keys of a transaction.

WAIT
Blocks until the asynchronous replication of all preceding write commands sent by the connection is completed.

WAITAOF
Blocks until all of the preceding write commands sent by the connection are written to the append-only file of the master and/or replicas.

WATCH
Monitors changes to keys to determine the execution of a transaction.

XACK
Returns the number of messages that were successfully acknowledged by the consumer group member of a stream.

XADD
Appends a new message to a stream. Creates the key if it doesn't exist.

XAUTOCLAIM
Changes, or acquires, ownership of messages in a consumer group, as if the messages were delivered to as consumer group member.

XCLAIM
Changes, or acquires, ownership of a message in a consumer group, as if the message was delivered a consumer group member.

XDEL
Returns the number of messages after removing them from a stream.

XGROUP CREATE
Creates a consumer group.

XGROUP CREATECONSUMER
Creates a consumer in a consumer group.

XGROUP DELCONSUMER
Deletes a consumer from a consumer group.

XGROUP DESTROY
Destroys a consumer group.

XGROUP SETID
Sets the last-delivered ID of a consumer group.

XINFO CONSUMERS
Returns a list of the consumers in a consumer group.

XINFO GROUPS
Returns a list of the consumer groups of a stream.

XINFO STREAM
Returns information about a stream.

XLEN
Return the number of messages in a stream.

XPENDING
Returns the information and entries from a stream consumer group's pending entries list.

XRANGE
Returns the messages from a stream within a range of IDs.

XREAD
Returns messages from multiple streams with IDs greater than the ones

XREADGROUP
Returns new or historical messages from a stream for a consumer in a group.

requested. Blocks until a message is available otherwise.

Blocks until a message is available otherwise.

XREVRANGE
Returns the messages from a stream within a range of IDs in reverse order.

XSETID
An internal command for replicating stream values.

XTRIM
Deletes messages from the beginning of a stream.

ZADD
Adds one or more members to a sorted set, or updates their scores. Creates the key if it doesn't exist.

ZCARD
Returns the number of members in a sorted set.

ZCOUNT
Returns the count of members in a sorted set that have scores within a range.

ZDIFF
Returns the difference between multiple sorted sets.

ZDIFFSTORE
Stores the difference of multiple sorted sets in a key.

ZINCRBY
Increments the score of a member in a sorted set.

ZINTER
Returns the intersect of multiple sorted sets.

ZINTERCARD
Returns the number of members of the intersect of multiple sorted sets.

ZINTERSTORE
Stores the intersect of multiple sorted sets in a key.

ZLEXCOUNT
Returns the number of members in a sorted set within a lexicographical range.

ZMPOP
Returns the highest- or lowest-scoring members from one or more sorted sets after removing them. Deletes the sorted set if the last member was popped.

ZMSCORE
Returns the score of one or more members in a sorted set.

ZPOPMAX
Returns the highest-scoring members from a sorted set after removing them. Deletes the sorted set if the last member was popped.

ZPOPMIN
Returns the lowest-scoring members from a sorted set after removing them. Deletes the sorted set if the last member was popped.

ZRANDMEMBER
Returns one or more random members from a sorted set.

ZRANGE
Returns members in a sorted set within a range of indexes.

ZRANGEBYLEX
Returns members in a sorted set within a lexicographical range.

ZRANGEBYSCORE
Returns members in a sorted set within a range of scores.

ZRANGESTORE
Stores a range of members from sorted set in a key.

ZRANK
Returns the index of a member in a sorted set ordered by ascending scores.

ZREM
Removes one or more members from a sorted set. Deletes the sorted set if all members were removed.

ZREMRANGEBYLEX
Removes members in a sorted set within a lexicographical range. Deletes the sorted set if all members were removed.

ZREMRANGEBYRANK
Removes members in a sorted set within a range of indexes. Deletes the sorted set if all members were removed.

ZREMRANGEBYSCORE
Removes members in a sorted set within a range of scores. Deletes the sorted set if all members were removed.

ZREVRANGE
Returns members in a sorted set within

ZREVRANGEBYLEX
Returns members in a sorted set within

ZREVRANGEBYSCORE
Returns members in a sorted set within

Returns members in a sorted set within a range of indexes in reverse order.

Returns members in a sorted set within a lexicographical range in reverse order.

Returns members in a sorted set within a range of scores in reverse order.

ZREVRANK
Returns the index of a member in a sorted set ordered by descending scores.

ZSCAN
Iterates over members and scores of a sorted set.

ZSCORE
Returns the score of a member in a sorted set.

ZUNION
Returns the union of multiple sorted sets.

ZUNIONSTORE
Stores the union of multiple sorted sets in a key.