Redis Database

What is Redis?

Redis is a very fast non-relational database that stores a mapping of keys to five different types of values. Redis supports in-memory persistent storage on disk, replication to scale read performance, and client-side sharding1 to scale write performance.

There are 2 types of commit we can use to avoid the data loss

1. **Trigger Mode :** The first method is a point-in-time dump either when certain conditions are met (a number of writes in a given period) or when one of the two dump-to-disk commands is called
2. **Append Mode :** The other method uses an append-only file that writes every command that alters data in Redis to disk as it happens. Depending on how careful you want to be with your data, append-only writing can be configured to never sync, sync once per second, or sync at the completion of every operation

Redis allows us to store keys that map to any one of five different data structure types; **STRING**s, **LIST**s, **SET**s, **HASH**es, and **ZSET**s. Each of the five different structures have some shared commands (DEL, TYPE, RENAME, and others), as well as some commands that can only be used by one or two of the structures.

Comparison between different database

Name	Type	Data storage options	Query types	Additional features
Redis	In-memory non-relational database	Strings, lists, sets, hashes, sorted sets	Commands for each data type for common access patterns, with bulk operations, and partial transaction support	Publish/Subscribe, master/slave replication, disk persistence, scripting (stored procedures)
memcached	In-memory key-value cache	Mapping of keys to values	Commands for create, read, update, delete, and a few others	Multithreaded server for additional performance
MySQL	Relational database	Databases of tables of rows, views over tables, spatial and third-party extensions	SELECT, INSERT, UPDATE, DELETE, functions, stored procedures	ACID compliant (with InnoDB), master /slave and master/master replication
PostgreSQL	Relational database	Databases of tables of rows, views over tables, spatial and third-party extensions, customizable types	SELECT, INSERT, UPDATE, DELETE, built-in functions, custom stored procedures	ACID compliant, master/slave replication, multi-master replication (third party)
MongoDB	On-disk non-relational document store	Databases of tables of schema-less BSON documents	Commands for create, read, update, delete, conditional queries, and more	Supports map-reduce operations, master /slave replication, sharding, spatial indexes

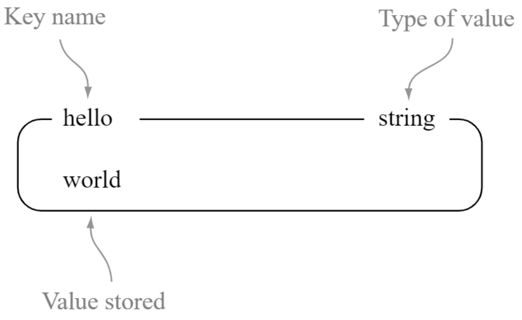
The five structures available in Redis

Structure type	What it contains	Structure read/write ability
STRING	Strings, integers, or floating-point values	Operate on the whole string, parts, increment/ decrement the integers and floats
LIST	Linked list of strings	Push or pop items from both ends, trim based on offsets, read individual or multiple items, find or remove items by value
SET	Unordered collection of unique strings	Add, fetch, or remove individual items, check membership, intersect, union, difference, fetch random items
HASH	Unordered hash table of keys to values	Add, fetch, or remove individual items, fetch the whole hash
ZSET (sorted set)	Ordered mapping of string members to floating-point scores, ordered by score	Add, fetch, or remove individual values, fetch items based on score ranges or member value

String Representation :

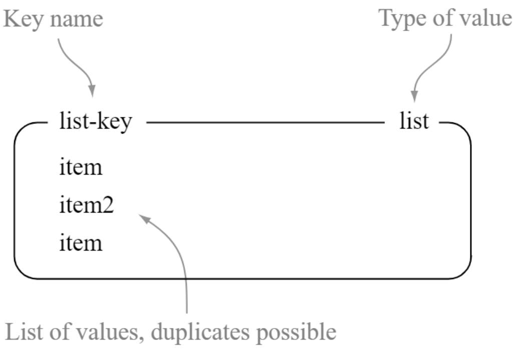
Commands used on **STRING** values

Command	What it does
GET	Fetches the data stored at the given key



SET	Sets the value stored at the given key
DEL	Deletes the value stored at the given key (works for all types)

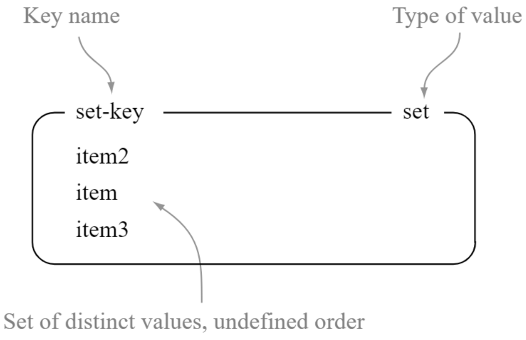
List Representation :



Commands used on **LIST** values

Command	What it does
RPUSH	Pushes the value onto the right end of the list
LRANGE	Fetches a range of values from the list
LINDEX	Fetches an item at a given position in the list
LPOP	Pops the value from the left end of the list and returns it

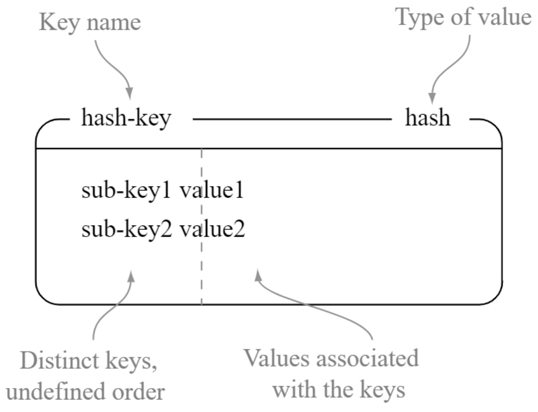
Set Representation :



Commands used on SET values

Command	What it does
SADD	Adds the item to the set
SMEMBERS	Returns the entire set of items
SISMEMBER	Checks if an item is in the set
SREM	Removes the item from the set, if it exists

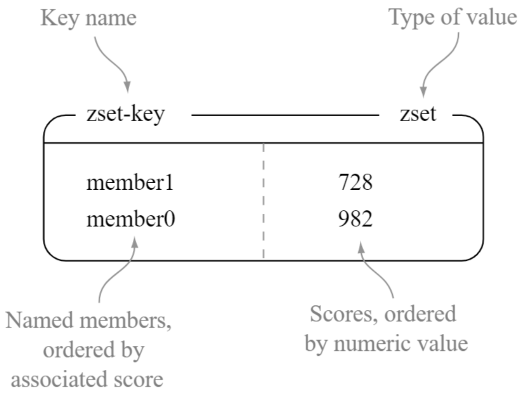
Hashes Representation :



Commands used on HASH values

Command	What it does
HSET	Stores the value at the key in the hash
HGET	Fetches the value at the given hash key
HGETALL	Fetches the entire hash
HDEL	Removes a key from the hash, if it exists

ZSET Representation :


Key name
Type of value
zset-key
zset
member1
728
member0
982
Named members, ordered by associated score
Scores, ordered by numeric value

Commands used on ZSET values

Command	What it does
ZADD	Adds member with the given score to the ZSET
ZRANGE	Fetches the items in the ZSET from their positions in sorted order
ZRANGEBYSCORE	Fetches items in the ZSET based on a range of scores
ZREM	Removes the item from the ZSET, if it exists