Key

ردیس یک دیتابیس nosql است که اطلاعات را به شکل کلید/مقدار ذخیره میکند. ردیس از انواع مختلفی از دیتاتایپ ها پشتیبانی میکند. ساده ترین نوع داده در ردیس string است. برای ایجاد کردن یک کلید میتوانید از دستور SET و برای خواندن اطلاعات یک کلید از دستور GET استفاده کنید.کلیدها در ردیس binary safe هستند یعنی میتوانند هرگونه دنباله باینری را ذخیره کنند. از یک string ساده گرفته تا محتویات یک عکس. همچنین stringهای خالی هم یک کلید معتبر هستند.

با دستور redis-cli وارد محیط redis می‌شویم

127.0.0.1:6379

===========================================================

string

stringها در ردیس ساده ترین نوع داده در ردیس هستند. stringها در ردیس binary safe هستند یعنی میتوانند هرگونه دنباله باینری را ذخیره کنند. از یک آبجکت زبان پایتون گرفته تا محتویات یک عکس. حداکثر حجم هر string میتواند 512 مگابایت باشد. تمامی اعداد در ردیس به عنوان string ذخیره میشوند. شما میتوانید از دستورات ردیس برای تغییر دادن stringها استفاده کنید.

===========================================================

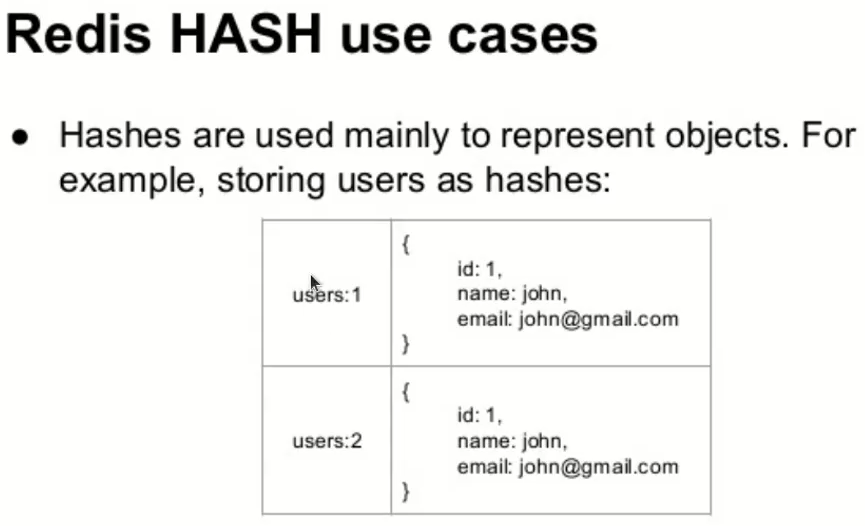
list

با استفاده از listها میتوانید چندین مقدار را داخل یک کلید ذخیره کنید. المان های لیست به ترتیب ورود طبقه بندی میشوند یعنی ترتیب ورود اطلاعات مهم است. هر لیست میتواند بیشتر از ۴ میلیارد المان در خود ذخیره کند. همچنین میتوانید با متدهایی که ردیس در اختیار شما قرار میدهد لیست های خود را تغییر بدهید.

===========================================================

hash

هش ها نوعی از داده هستند که میتوانید در یک کلید چندین مقدار به شکل کلید/مقدار ذخیره کنید. در ردیس هش ها توانایی ذخیره بیش از ۴ میلیادر المان را دارند. هش هایی با بیش از هزار فیلد به شکلی ذخیره میشوند که فضای کمی را اشغال کنند پس شما میتوانید جندین میلیون رکورد را بدون مشکل ذخیره کنید. بیشترین استفاده از هش ها در redis برای دخیره آبجکت ها است



دستورات:

HSET , HGET , HSETNX , HMSET , HDEL , DEL , HEXIST ,

HGETALL , HINCRBY , HINCRBYFLOAT , HKEYS , HVALS , HLEN ,

HMGET , HSTRLEN ,

===========================================================

set در ردیس مجموعه ای از المان هایی است که به شکل نامنظم در یک کلید ذخیره میشوند. با استفاده از متدهایی که redis در اختیار ما قرار داده امکان اضافه کردن و حذف کردن المان ها وجود دارد. setها در ردیس اجازه ذخیره المان های تکراری را نمیدهد و اضافه کردن چندباره المان ها تغییری در تعداد نخواهد داشت. همچنین ردیس از عملیات ها ریاضی مثل اجتماع، اشتراک و تفاوت ها پشتیبانی میکند. هر set امکان ذخیره بیش از ۴ میلیارد المان را دارد

دستورات :

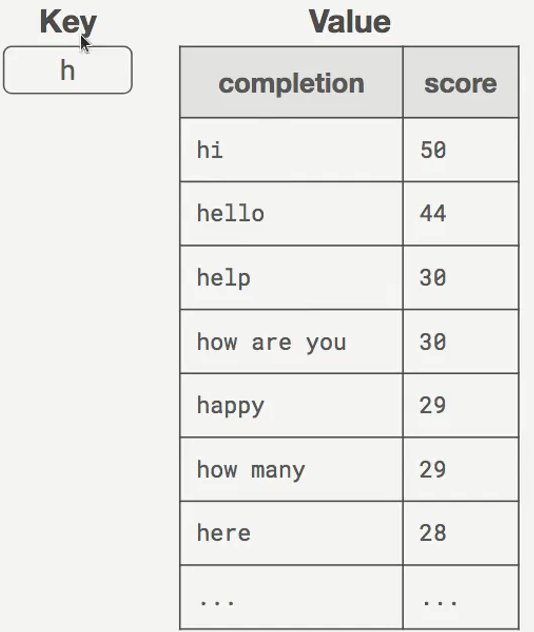
SADD , SMEMBERS , SCARD , SISMEMBER , SPOP , SRANDMEMBER ,

SREM , SMOVE , SDIFF , SINTER , SUNION , SDIFFSTORE

===========================================================

نوع داده sorted set دقیقا مشابه setها است و نمیتوان المان های تکراری در آن ذخیره کرد. تنها تفاوتی که این دو نوع با هم دارند اینست که المان ها در

sorted set یک score یا امتیاز دارند که ردیس المان ها را به ترتیب این scoreها ذخیره میکند. در این نوع داده المان ها نمیتوانند موارد تکراری داشته باشند اما امتیازها میتوانند تکرار شوند. در sorted setها اعمال خواندن، اضافه کردن، حذف کردن و آپدیت کردن المان ها بسیار سریع است



دستورات:

ZADD:ساخت , ZRANGE:خواندن , ZCARD: تعداد عضوها ,

ZCOUNT : تعداد عضوها در رنج , ZINCRBY : اضافه کردن مقداری

ZINTERSTOR : ذخیره کردن مشترک ها , ZPOPMAX , ZPOPMIN ,

ZRANK: چندمین عضو بودن , ZREM:حذف ,

===========================================================

hyperLogLog ها دقیقا مثل set ها هستند یعنی امکان ذخیره موارد تکراری وجود ندارد و اینکه ترتیب ورود اطلاعات مهم نیست. این ساختار برای زمانی مناسب است که خود اطلاعات اهمیتی ندارند و فقط تعداد آنها مهم است. مثلا برنامه ای که به صورت real-time نیاز به ذخیره تعداد کاربرهای unique دارد میتواند از این نوع داده استفاده کند. hyperLogLogها در ردیس بسیار بهینه تر از set ها هستند

دستورات:

PFADD:ساخت , PFCOUNT: تعداد , PFMERGE: ترکیب چندتا و ذخیره

===========================================================

pub/sub یک سیستم پیغام رسانی است که در آن فرستنده (publisher) یک پیغام را در یک کانال(channel) فرستاده و گیرنده(subscriber) با متصل شدن به کانال میتواند پیغام ها را خوانده و پردازش کند. در ردیس گیرنده میتواند به چندین کانال وصل شود و محدودیتی وجود ندارد. در این سیستم فرستنده و گیرنده کاملا از هم جدا(decouple) شده اند. فرستنده ها هیچ اطلاعی از گیرنده ها ندارد و بالعکس گیرنده ها هیچ اطلاعی از فرستنده ها ندارند. این جداسازی باعث شده که سرعت ارسال اطلاعات بیشتر شده و فرستنده منتظر پردازش ها گیرنده نباشد.