Semantix

# Redis

Aula 3

# Quem sou eu?

## Eu sou Rodrigo Augusto Rebouças.

Engenheiro de dados da Semantix
Instrutor do Semantix Mentoring Academy

Você pode me encontrar em:
rodrigo.augusto@semantix.com.br

# Sets

Semantix

# Redis Sets

- Sets são coleções não ordenadas de strings
  - Expressar relações entre objetos
- Sintaxe
  - Adicionar elementos
    - sadd <chave> <valor1> ... <valorN>
  - Retornar todos os elementos
    - smembers <chave>
  - Recuperar um elemento e remove-lo do set
    - spop <chave>

```
127.0.0.1:6379> sadd views:nome Rodrigo Carlos Ana Augusto
(integer) 4
127.0.0.1:6379> smembers views:nome
1) "Augusto"
2) "Ana"
3) "Carlos"
4) "Rodrigo"
127.0.0.1:6379> spop views:nome
"Carlos"
```

# Redis Sets

- Verificar se um elemento existe
  - Sintaxe: sismember <chave> <valor>
- Visualizar o número de elementos
  - Sintaxe: scard <chave>
- Remover um elemento
  - Sintaxe: srem <chave>

```
127.0.0.1:6379> smembers views:nome
1) "Augusto"
2) "Ana"
3) "Rodrigo"
127.0.0.1:6379> sismember views:nome Rodrigo
(integer) 1
127.0.0.1:6379> scard views:nome
(integer) 3
127.0.0.1:6379> srem views:nome Augusto
(integer) 1
127.0.0.1:6379> smembers views:nome
1) "Ana"
2) "Rodrigo"
```

# Redis Múltiplos Sets

- Interseção de vários sets
  - Sintaxe: sinter <chave1> ... <chaveN>
- Diferença de vários sets
  - Sintaxe: sdiff <chave1> ... <chaveN>
- União de vários sets
  - Sintaxe: sunion <chave1> ... <chaveN>

```
127.0.0.1:6379> smembers views:nome
1) "Ana"
2) "Rodrigo"
127.0.0.1:6379> sadd comentario:nome Rodrigo Maria
(integer) 2
127.0.0.1:6379> sinter views:nome comentario:nome
1) "Rodrigo"
127.0.0.1:6379> sdiff views:nome comentario:nome
1) "Ana"
127.0.0.1:6379> sunion views:nome comentario:nome
1) "Ana"
2) "Maria"
3) "Rodrigo"
```

- Todos os comandos tem a versão para armazenar em uma nova chave

# Redis Múltiplos Sets

- Armazenar os múltiplos sets em  outra chave
  - Sintaxe
    - sinterstore <chaveArmazenamento> <chave1> ... <chaveN>
    - sdiffstore <chaveArmazenamento> <chave1> ... <chaveN>
    - sunionstore <chaveArmazenamento> <chave1> ... <chaveN>

```
127.0.0.1:6379> sinterstore inter_views_comentario:nome views:nome comentario:nome
(integer) 1
127.0.0.1:6379> sdiffstore diff_views_comentario:nome views:nome comentario:nome
(integer) 1
127.0.0.1:6379> sunionstore union_views_comentario:nome views:nome comentario:nome
(integer) 3
127.0.0.1:6379> smembers union_views_comentario:nome
1) "Ana"
2) "Maria"
3) "Rodrigo"
```

# Exercícios - Sets

1. Deletar a chave “pesquisa:produto”
2. Criar a chave "pesquisa:produto" do tipo set com os seguintes valores: monitor, mouse e teclado
3. Visualizar a quantidade de valores da chave
4. Retornar todos os elementos da chave
5. Verificar se existe o valor monitor
6. Remover o valor monitor
7. Recuperar um elemento e remove-lo do set
8. Criar a chave "pesquisa:desconto“ do tipo set com os seguintes valores: memória RAM, monitor, teclado, HD
9. Próximas questões fazem uso dos sets pesquisa:produto e pesquisa:desconto
   - Visualizar a interseção entre os 2 sets
   - Visualizar a diferença entre os 2 sets
   - Criar o set "pesquisa:produto_desconto" com a união entre os 2 sets

# Sets Ordenados

Semantix

# Redis Sets Ordenados

- *Sorted sets* são compostos de elementos de string únicos e não repetitivos
  - Combinação de Set e Hash
  - Também é um Set
- Cada elemento é associado a um score
  - Semelhante ao Hash
- Sintaxe
  - Adicionar elementos
    - zadd <chave> <score1> <valor1> ... <scoreN> <valorN>
  - Visualizar elementos em um intervalo na lista
    - Crescente - zrange <chave> <inicio> <fim> [withscores]
    - Decrescente - zrevrange <chave> <inicio> <fim> [withscores]

# Redis Sets Ordenados

- Ex.

```
127.0.0.1:6379> zadd views:nomes 10 Rodrigo 15 Carlos 10 Ana
(integer) 3
127.0.0.1:6379> zrange views:nomes 0 -1
1) "Ana"
2) "Rodrigo"
3) "Carlos"
127.0.0.1:6379> zrevrange views:nomes 0 -1
1) "Carlos"
2) "Rodrigo"
3) "Ana"
127.0.0.1:6379> zrange views:nomes 0 -1 withscores
1) "Ana"
2) "10"
3) "Rodrigo"
4) "10"
5) "Carlos"
6) "15"
```

# Redis Sets Ordenados

- Recuperar um elemento e remove-lo do set
- Sintaxe
  - Maior score - zpopmax <chave>
  - Menor score - zpopmin <chave>
- Bloquear se o set estiver vazio até um determinado tempo *t*
  - Sintaxe
    - Maior score - bzpopmax <chave> <t>
    - Menor score - bzpopmin <chave> <t>

```
127.0.0.1:6379> zrange views:nomes 0 -1 withscores
1) "Ana"
2) "10"
3) "Rodrigo"
4) "10"
5) "Carlos"
6) "15"
127.0.0.1:6379> zpopmax views:nomes
1) "Carlos"
2) "15"
127.0.0.1:6379> zpopmin views:nomes
1) "Ana"
2) "10"
127.0.0.1:6379> bzpopmin views:nomes 5
1) "views:nomes"
2) "Rodrigo"
3) "10"
127.0.0.1:6379> bzpopmin views:nomes 5
(nil)
(5.08s)
127.0.0.1:6379>
```

# Redis Sets Ordenados

- Visualizar a  posição de um elemento
  - Sintaxe
    - zrank <chave> <valor>
    - zrevrank <chave> <valor>
- Visualizar o score de um elemento
  - Sintaxe: zscore <chave> <valor>
- Visualizar o número de elementos
  - Sintaxe: zcard <chave>
- Remover um elemento específico
  - Sintaxe: zrem <chave> <valor>

```
127.0.0.1:6379> zrange views:nomes 0 -1 withscores
1) "Ana"
2) "10"
3) "Rodrigo"
4) "10"
5) "Carlos"
6) "15"
127.0.0.1:6379> zrank views:nomes Rodrigo
(integer) 1
127.0.0.1:6379> zscore views:nomes Rodrigo
"10"
127.0.0.1:6379> zcard views:nomes
(integer) 3
127.0.0.1:6379> zrem views:nomes Carlos
(integer) 1
127.0.0.1:6379> zcard views:nomes
(integer) 2
```

# Exercícios – Sets Ordenados

1. Deletar a chave "pesquisa:produto"
2. Criar a chave "pesquisa:produto" do tipo set ordenado com os seguintes valores:
   - Valor: monitor, Score: 100
   - Valor: HD, Score: 20
   - Valor: mouse, Score: 10
   - Valor: teclado, Score: 50

O score representa a quantidade de pesquisas feitas para aquele produto na aplicação

1. Visualizar a quantidade de produtos
2. Visualizar todos os produtos do mais pesquisado ao menos pesquisado
3. Visualizar o rank do produto teclado
4. Visualizar o score do produto teclado
5. Remover o valor HD da chave
6. Recuperar e remover do set o produto mais pesquisado
7. Recuperar e remover do set o produto menos pesquisado
8. Visualizar todos os produtos

# Hashes

Semantix

# Redis Hashes

- Hashes são pares de valor de campo
  - Representar objetos
- Sintaxe
  - Definir o valor de um campo de hash
    - hmset <chave> <campo1> <valor> ...
  - Obter o valor de um campo de hash
    - hget <chave> <campo>
  - Obter os valores dos campos de hash
    - hmget <chave> <campo1> ...
  - Obter todos os campos e valores de uma hash
    - hgetall <chave>

```
127.0.0.1:6379> hset produto:100 nome mouse qtd 10
(integer) 2
127.0.0.1:6379> hget produto:100 nome
"mouse"
127.0.0.1:6379> hmget produto:100 nome qtd
1) "mouse"
2) "10"
127.0.0.1:6379> hgetall produto:100
1) "nome"
2) "mouse"
3) "qtd"
4) "10"
```

# Redis Hashes

- Incrementar valores nos campos
  - Sintaxe
    - hincrby <chave> <campo> <incremento>

```
127.0.0.1:6379> hget produto:100 qtd
"10"
127.0.0.1:6379> hincrby produto:100 qtd 1
(integer) 11
127.0.0.1:6379> hincrby produto:100 qtd -1
(integer) 10
127.0.0.1:6379> hincrby produto:100 qtd 15
(integer) 25
```

# Redis Hashes

- Obter o número de campos
  - Sintaxe: hlen <chave>
- Obter o tamanho do valor de um campo
  - Sintaxe: hstrlen <chave> <campo>
- Obter todos os campos da hash
  - Sintaxe: hkey <chave> <campo>
- Obter todos os valores da hash
  - Sintaxe: hvals <chave>
- Deletar o campo
  - Sintaxe: hdel <chave> <campo>

```
127.0.0.1:6379> hlen produto:100
(integer) 2
127.0.0.1:6379> hstrlen produto:100 nome
(integer) 5
127.0.0.1:6379> hkeys produto:100
1) "nome"
2) "qtd"
127.0.0.1:6379> hvals produto:100
1) "mouse"
2) "10"
127.0.0.1:6379> hdel produto:100 nome
(integer) 1
127.0.0.1:6379> hlen produto:100
(integer) 1
```

# Exercícios

1. Deletar a chave “usuario:100”
2. Criar uma chave “usuario:100” do tipo hash com a seguinte estrutura
   - nome – Augusto
   - estado – SP
   - views – 10
3. Visualizar todas as chaves e valores
4. Contar a quantidade de campos
5. Visualizar apenas o nome e views
6. Contar o tamanho do valor do campo nome
7. Incrementar em 2 o valor do campo views
8. Visualizar apenas os campos
9. Visualizar apenas os valores
10. Deletar o campo estado

Semantix

# Obrigado!

Alguma pergunta?


Você pode me encontrar em:

rodrigo.augusto@semantix.com.br

GET SMARTER