





Banco NoSQL - Redis

👤 Person	 Ygor Nogueira  Matheus Costa
⚙️ Status	In Progress

Ele vai conter exemplos práticos das operações CRUD do mesmo banco de dados dos filmes

Introdução:

O Redis é um banco de dados NoSQL baseado na relação "chave-valor", onde as chaves e os valores são armazenados como strings. Como ele não trabalha com esquemas relacionais da mesma forma que o MySQL, vamos usar `hashes` para armazenar as informações e relacionar as chaves entre si. Então, utilizando o Redis Cloud via AWS, através do CLI nos populamos o banco utilizando as propriedades indicadas.

Vale ressaltar alguns pontos sobre o Redis:

1. Já que o Redis não tem suporte nativo para relacionamentos como um banco relacional (chaves estrangeiras e afins), foram utilizadas chaves compostas para relacionar a tabela `exibicao`.
2. Como o Redis armazena tudo que o compõe (como sets, hashes, listas, sorted sets) internamente em strings, até mesmo os números, ele acaba se tornando muito eficiente em consultas.
3. Portanto, ele tem muita serventia em serviços web por exemplo, que consultam muito cache temporário para ficar retornando valores. Com ele, estas interações podem ser reduzidas drasticamente em tempo de operação, chegando a ter retornos de milissegundos.


Criação da estrutura "Programações de Filmes" no Redis:


- **canal** será armazenado em uma `hash`, onde a chave será `canal:{num_canal}` e o valor será um `hash` com `nome` e `sigla`.
- **filme** será armazenado da mesma forma, com a chave `filme:{num_filme}` e o valor com os atributos `titulo_original`, `titulo_brasil`, `ano_lancamento`, `pais_origem`, `categoria` e `duracao`.
- **exibicao** será armazenado em uma chave composta de `exibicao:{num_filme}:{num_canal}:{data}`, onde os atributos serão `num_filme`, `num_canal` e `data`.

CRUD utilizando o Redis Insight

No **Redis Insight**, temos uma interface gráfica para realizar essas operações CRUD de forma mais intuitiva:

- **Create:** No Redis Insight, você pode clicar em "Add New Key" e preencher as informações necessárias para criar hashes ou outros tipos de dados.



 Bulk Actions

+ Key

New Key

Key Type*

● Hash




▼

TTL

No limit

Key Name*

canal:9

num_canal	9	
nome	Disney Channel	
sigla	DIS	

+

Cancel

Add Key

- **Read:** Navegue até as chaves criadas e clique sobre elas para ver os detalhes.

HASH canal:2

112 B

Length: 3

TTL: No limit

< 1 min

↺

↻

+

🗑

Field	Value	
num_canal	2	🗑
nome	Home Box Office	🗑
sigla	HBO	🗑

- **Update:** Selecione uma chave e clique em "Edit" para atualizar os valores diretamente pela interface.

HASH canal:9

112 B

Length: 3

TTL: No limit

< 1 min

↺

↻

+

🗑

Field	Value	
num_canal	9	🗑
nome	Disney Channel	✎ 🗑
sigla	DIS	🗑

HASH canal:9

112 B Length: 3 TTL: No limit < 1 min

Field	Value
num_canal	9
nome	Disney Channel
sigla	DIS

- **Delete:** Clique em uma chave e selecione "Delete" para removê-la.

HASH canal:9

112 B Length: 3 TTL: No limit < 1 min

canal:9 will be deleted.

Delete

Field	Value
num_canal	9
nome	Disney Channel
sigla	DIS

Alternativamente, para realizar todas as operações, podemos também realizá-las via comandos diretamente no terminal:

- Create: `HMSET key fieldvalue [field value ...]` podendo criar qualquer valor ou valores e atribuí-los

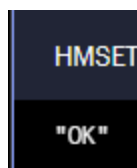
Exemplos:

Inserir novos registros em canal, filme e exibicao

Para inserir um novo canal: `HMSET canal:6 num_canal 6 nome "Discovery Channel" sigla "DC"`

Para inserir um novo filme: `HMSET filme:13 num_filme 13 titulo_original "Barbie" titulo_brasil "Barbie" ano_lancamento 2023 pais_origem "Estados Unidos" categoria "Comédia" duracao 114`

Para inserir uma nova exibição (no slot 11 que ainda não tem registro) para o filme número 13 (Barbie) que acabamos de criar, no canal número 6 (DC) que também acabamos de criar: `HMSET exibicao:11:6:2024-09-01_18:00 num_filme 13 num_canal 6 data "2024-09-01 18:00:00"`



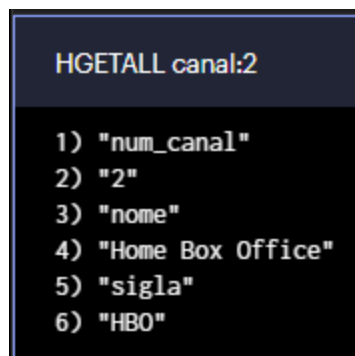
ao final de qualquer inserção, caso todos os dados estejam corretos, ele sempre retorna "OK"

- Read: `HGETALL key`

Exemplos:

Para buscar um canal com

`num_canal = 2: HGETALL canal:2`



Para buscar um filme com `num_filme = 2: HGETALL filme:2`

```
HGETALL filme:2
```

```
1) "num_filme"  
2) "2"  
3) "titulo_original"  
4) "Interstellar"  
5) "titulo_brasil"  
6) "Interestelar"  
7) "ano_lancamento"  
8) "2014"  
9) "pais_origem"  
10) "Estados Unidos"  
11) "categoria"  
12) "Fic\xc3\xa7\xc3\xa3o cient\xadfica"  
13) "duracao"  
14) "169"
```

Para buscar uma exibição que ocorreu no canal 2 para o filme 1 em uma data específica: `HGETALL exibicao:1:2:2024-08-30_15:25`

```
1 Command(s) - 1 success, 0 error(s)
```

```
> HGETALL exibicao:1:2:2024-08-30_15:25  
1) "num_filme"  
2) "1"  
3) "num_canal"  
4) "2"  
5) "data"  
6) "2024-08-30 15:25:00"
```

- Update: `HSET key fieldvalue [field value ...]` podendo alterar qualquer valor ou valores já atribuídos

Exemplos:

Se quiser atualizar o nome do canal com

`num_canal = 2` para "HBO Max": `HSET canal:2 nome "HBO Max"`

```
1 Command(s) - 1 success, 0 error(s)

> HSET canal:2 nome "HBO Max"
(integer) 0
```

Para atualizar o título brasileiro de

`filme:2` para "Interestelar - Viagem no Espaço": `HSET filme:2 titulo_brasil "Interestelar - Viagem no Espaço"`

```
1 Command(s) - 1 success, 0 error(s)

> HSET filme:2 titulo_brasil "Interestelar - Viagem no Espaço"
(integer) 0
```

Para mudar a data de exibição do filme

`filme:1` no canal `canal:2` para "2024-08-31 17:00:00": `HSET exibicao:1:2:2024-08-30_15:25 data "2024-08-31 17:00:00"`

```
1 Command(s) - 1 success, 0 error(s)

> HSET exibicao:1:2:2024-08-30_15:25 data "2024-08-31 17:00:00"
(integer) 0
```

- Delete: `DEL key[key ...]`

Exemplos:

Para remover o canal

`canal:6`: `DEL canal:6`


```
1 Command(s) - 1 success, 0 error(s)
```

```
> DEL canal:6  
(integer) 1
```

Para remover o filme

```
filme:11 : DEL filme:11
```

```
1 Command(s) - 1 success, 0 error(s)
```

```
> DEL filme:11  
(integer) 1
```

Para remover uma exibição de

```
filme:1 no canal:2 em uma data específica: DEL exibicao:1:2:2024-08-30_15:25
```

```
1 Command(s) - 1 success, 0 error(s)
```

```
> DEL exibicao:1:2:2024-08-30_15:25  
(integer) 1
```