

Una vez instalado y cargado el módulo de **RediSearch**, ya podemos empezar a describir detalladamente los componentes usados por el motor de búsqueda. Comenzamos con los índices.

Iniciamos la lección introduciendo el concepto de índice de búsqueda. Y a continuación, presentamos los comandos con los que crear, suprimir y consultar sus metadatos.

Al finalizar la lección, sabrá:

- Qué es un índice de búsqueda.
- Para qué sirve un índice.
- Cómo crear y suprimir índices.
- Cómo consultar los metadatos de los índices.

Introducción

Un **índice de búsqueda** (*search index*) es una estructura de datos, mantenida por el motor de búsqueda, en la que se registra los términos que aparecen en un conjunto de documentos. Los motores de búsqueda suelen utilizar **índices invertidos** (*inverted indices*), los cuales catalogan los distintos términos que aparecen en los documentos indexados, para cada uno de los cuales registra:

- En qué documentos aparece.
- En qué campos aparece.
- En qué posiciones dentro de los campos aparece.

Así resulta muy fácil encontrar los documentos que contienen uno o más términos.

Una base de datos **Redis** puede tener tantos índices de búsqueda como sea necesario. Generalmente, cada uno de ellos indexando un determinado conjunto de documentos relacionados con un determinado aspecto o funcionalidad. A su vez, los documentos a indexar pueden encontrarse en uno o varios índices si es necesario.

Esquemas de los índices

Todo índice presenta un **esquema** (*schema*), conjunto de campos de los documentos que debe indexar. Sólo se extraerá los términos de los campos indicados en el esquema. El resto se omitirán. A la hora de crear el índice, hay que indicar su esquema.

Tal como veremos en la siguiente lección, un documento no es más que un par clave-valor de la base de datos, cuyo valor es de tipo array asociativo (*hash*). Recordemos que este tipo no es más que un registro de datos que presenta campos, cada uno de los cuales debe tener un valor de tipo cadena. Las cuales pueden representar textos o números. Cuando indiquemos el esquema, hay que indicarle al índice exactamente qué tipo de valor tiene: un texto, un número o una coordenada geoespacial. Esto le ayuda a indexarlo mejor atendiendo a su contenido.

Peso del campo

En un documento, de cara a las búsquedas, cada campo puede tener un **peso** (*weight*), esto es, un nivel de importancia. Por ejemplo, si estamos indexando un catálogo de productos, un término en el campo nombre suele tener mayor peso que en la descripción. Así, cuando se busca un término, digamos groso modo, para cada documento se extraerá la puntuación total del término en el documento según su número de apariciones y el peso de cada aparición.

Este peso se puede especificar en los campos de tipo texto. Indique el peso para cada campo de este tipo según su nivel de importancia para las consultas.

Creación de índices

La **creación de un índice** (*index creation*) es la operación mediante la cual se crea un índice de búsqueda en **Redis**. Hay que utilizar el comando **FT.CREATE**, cuya sintaxis es como sigue:

FT.CREATE nombre SCHEMA campos

FT.CREATE nombre parámetros SCHEMA campos

El nombre del índice lo identifica del resto y se usa en las consultas para añadirle documentos y realizar búsquedas en sus documentos indexados. Si ya existe un índice con ese nombre, fallará la operación.

Recordemos que todo índice tiene asociado un esquema, el conjunto de campos a indexar de los documentos. Cualquier campo del documento que no se encuentre en este esquema, no se indexará. La sintaxis de los campos del esquema es:

campo TEXT

campo TEXT WEIGHT peso

campo NUMERIC

campo GEO

Los campos indexables deben ser de tipo texto, numérico o coordenada geográfica. En caso de ser textual, se recomienda encarecidamente indicar su peso, cuyo valor predeterminado es **1.0**.

Veamos un ejemplo ilustrativo de un índice con un esquema formado por dos campos, name y origin:

```
127.0.0.1:6379> FT.CREATE bands SCHEMA name TEXT WEIGHT 10 origin TEXT WEIGHT 5
OK
127.0.0.1:6379>
```

Opciones de almacenamiento

Recordemos que cada vez que añadimos un nuevo documento a un índice, éste lo analiza y extrae sus términos que es finalmente lo que indexa. De cada término, almacena el campo y el documento en el que aparece, así como su posición. Si lo deseamos, podemos omitir esta información, reduciendo así la memoria consumida, pero sacrificando las consultas de búsqueda.

Para este fin, disponemos de los siguientes parámetros a ubicar antes del esquema:

- **NOFIELDS**. No registrar los campos en que aparece cada término. Sólo mantener el documento en el que aparece.
Su uso, impide la aplicación de filtros por campo en las consultas de búsqueda.
- **NOOFFSETS**. No almacenar la posición en la que aparece el término.
No podrá realizar consultas de búsqueda exactas.

Ejemplo:

```
127.0.0.1:6379> FT.CREATE bands NOOFFSETS SCHEMA name TEXT origin TEXT
OK
127.0.0.1:6379>
```

Supresión de índice

La **supresión de índice** (*index drop*) es la operación mediante la cual se elimina un índice de la base de datos y todo su contenido. Se realiza mediante el comando **FT.DROP**:

FT.DROP índice

Ojo, suprimir un índice no suprime los documentos de la base de datos también. Sólo el índice y sus datos internos.

Consulta de metadatos del índice

Para consultar la definición de un índice y algunas estadísticas como, por ejemplo, el número de documentos, el número de términos distintos y el tamaño promedio de los documentos, se puede utilizar el comando **FT.INFO**:

FT.INFO índice

A continuación, se muestra un ejemplo ilustrativo:

```
127.0.0.1:6379> FT.INFO bands
```

```
1) index_name
2) bands
3) fields
4) 1) 1) name
    2) type
    3) TEXT
    4) weight
    5) "10"
   2) 1) origin
    2) type
    3) TEXT
    4) weight
    5) "5"
5) num_docs
6) "8"
7) max_doc_id
8) "8"
9) num_terms
10) "21"
11) num_records
12) "26"
13) inverted_sz_mb
14) "0.000148773193359375"
15) inverted_cap_mb
16) "0"
17) inverted_cap_ovh
18) "inf"
19) offset_vectors_sz_mb
20) "2.47955322265625e-05"
21) skip_index_size_mb
22) "0"
23) score_index_size_mb
24) "0"
25) doc_table_size_mb
26) "0.00041866302490234375"
27) key_table_size_mb
28) "0.00039196014404296875"
29) records_per_doc_avg
30) "3.25"
31) bytes_per_record_avg
32) "6"
33) offsets_per_term_avg
34) "1"
35) offset_bits_per_record_avg
36) "8"
127.0.0.1:6379>
```