

Una vez visto cómo crear índices y añadirles documentos, ha llegado el momento de presentar las consultas de búsqueda. Lo que usamos para determinar qué documentos presentan uno o más términos.

Se abre la lección introduciendo y clasificando las consultas de búsqueda. Después, se describe cómo realizar consultas simples y, a continuación, compuestas. Seguimos con las palabras vacías, aquellas que no se indexan nunca. Finalmente, cómo combinar consultas, realizar filtros y devolver resultados sin contenido.

Al finalizar la lección, sabrá:

- Qué es y cómo realizar consultas de búsqueda sobre índices.
- Qué es una palabra vacía.
- Cómo combinar varios criterios de búsqueda en una misma consulta.
- Cómo definir filtros.
- Cómo obtener resultados formados sólo por las claves de los documentos.

Introducción

Una **consulta de búsqueda** (*search query*) es aquella que examina un índice para extraer aquellos documentos que contienen uno o más términos especificados por el usuario. Atendiendo a lo que se busca, se distingue entre búsquedas simples o compuestas.

Una **búsqueda simple** (*simple search*) es aquella que consulta un único término. Mientras que una **búsqueda compuesta** (*composite search*), varios.

En **Redisearch**, las consultas de búsqueda, independientemente de su tipo, se realizan con el comando **FT.SEARCH**, cuya sintaxis básica es:

```
FT.SEARCH índice criteriosDeBúsqueda
FT.SEARCH índice criteriosDeBúsqueda parámetros
```

Búsquedas simples

Tal como acabamos de ver, una **búsqueda simple** (*simple search*) o **búsqueda de término** (*term search*) es aquella que consulta un único término. La sintaxis básica de **FT.SEARCH** para este tipo de búsquedas es como sigue:

```
FT.SEARCH índice término
FT.SEARCH índice término parámetros
```

A continuación, un ejemplo ilustrativo que lista aquellos documentos que contienen el término **UK**:

```
127.0.0.1:6379> FT.SEARCH bands UK
1) (integer) 2
2) "band:athlete:en"
3) 1) "name"
   2) "Athlete"
   3) "origin"
   4) "UK"
4) "band:joy-division:en"
5) 1) "name"
   2) "Joy Division"
   3) "origin"
   4) "UK"
127.0.0.1:6379>
```

Observe que no hace falta indicar en qué campos buscar. Lo hace en todos. Más adelante, mostramos cómo restringir las consultas a determinados campos.

Búsquedas compuestas

Por su parte, una **búsqueda compuesta** (*composite search*) es aquella que extrae aquellos documentos que presentan dos o más términos dados. En estos casos hay que delimitar los términos entre comillas, recomendándose las simples (').

Búsquedas de frase

Un tipo de búsqueda compuesta es la **búsqueda de frase exacta** (*exact phrase search*), aquella que busca una determinada secuencia de términos en un determinado orden. Para ello, hay que delimitar la frase entre comillas dobles ("), además de las que ya hay que indicar por especificar varios términos.

Ejemplo:

```
127.0.0.1:6379> FT.SEARCH bands '"motorcycle club"'
```

- 1) (integer) 1
- 2) "band:black-rebel-motorcycle-club:en"
- 3) 1) "name"
- 2) "Black Rebel Motorcycle Club"
- 3) "origin"
- 4) "US"

```
127.0.0.1:6379>
```

Si el orden de los términos no importa, es decir, deseamos que aparezcan *todos* los términos indicados sin importar si lo hacen adyacentemente ni en que orden, lo que estamos realizando es una **búsqueda de frase simple** (*simple phrase search*). Basta con indicar los términos, tal como se ilustra a continuación:

```
127.0.0.1:6379> FT.SEARCH bands 'club motorcycle'
```

- 1) (integer) 1
- 2) "band:black-rebel-motorcycle-club:en"
- 3) 1) "name"
- 2) "Black Rebel Motorcycle Club"
- 3) "origin"
- 4) "US"

```
127.0.0.1:6379> FT.SEARCH bands '"club motorcycle"'
```

- 1) (integer) 0

```
127.0.0.1:6379>
```

Búsqueda de unión

Por otra parte, tenemos la **búsqueda de unión** (*union search*), aquella que requiere que aparezca al menos uno de los términos indicados. Los términos se separan por |:

```
127.0.0.1:6379> FT.SEARCH bands 'club|fanclub'
```

- 1) (integer) 3
- 2) "band:teenage-fanclub:en"
- 3) 1) "name"
- 2) "Teenage Fanclub"
- 3) "origin"
- 4) "UK"
- 4) "band:bombay-bicycle-club:en"
- 5) 1) "name"
- 2) "Bombay Bicycle Club"
- 3) "origin"
- 4) "UK"
- 6) "band:black-rebel-motorcycle-club:en"
- 7) 1) "name"
- 2) "Black Rebel Motorcycle Club"
- 3) "origin"
- 4) "US"

```
27.0.0.1:6379>
```

Palabras vacías

Una **palabra vacía** (*stop word*) es aquella que no suele aportar nada al significado. Por ejemplo, las siguientes forman la lista de palabras vacías inglesas predeterminadas usadas por **RediSearch**: *a, is, the, an, and, are, as, at, be, but, by, if, in, into, it, no, not, of, on, or, such, that, their, then, there, these, they, this,*

to, *was*, *will* y *with*. Se consideran como poco aportadoras al significado del contenido procesado porque son muy utilizadas o muy cortas y no ayudan al procesador del lenguaje para extraer información relevante.

Por lo general, las palabras vacías no se indexan. Lo que puede dificultar la búsqueda exacta si aparece en la frase a buscar. En **Redisearch**, cuando nuestra consulta incluye una palabra vacía, se genera un error sintáctico:

```
127.0.0.1:6379> FT.SEARCH bands and
(error) Syntax error at offset 0 near ''
127.0.0.1:6379>
```

Para resolver el problema, hay que utilizar el parámetro **NOSTOPWORDS**:

```
127.0.0.1:6379> FT.SEARCH bands and NOSTOPWORDS
1) (integer) 0
127.0.0.1:6379>
```

En este caso, ya no se propaga ningún error, pero como el término a buscar es una palabra vacía y éstas no se indexan, el resultado es vacío. No olvide que las palabras vacías no se indexan.

Combinación de búsquedas

La **combinación de búsquedas** (*search combination*) consiste en indicar varias consultas de términos simples, frases y/o uniones. No hay ningún problema en hacerlo. Eso sí, no hay que olvidar delimitar la consulta por comillas simples y recordar que:

- El espacio indica que los criterios que le rodean deben aparecer en cualquier orden, como los operadores **&&** de **JavaScript** y **and** de **Lua**.
- El carácter **|** denota uno, otro o ambos, como los operadores **||** de **JavaScript** y **or** de **Lua**.
- Para realizar una búsqueda de frase exacta, ésta se debe delimitar por comillas dobles.
- Se puede usar los paréntesis para agrupar búsquedas.

Filtros

Un **filtro** (*filter*) permite restringir las búsquedas a sólo determinados documentos o campos del índice.

Filtros de campos

Un **filtro de campo** (*field filter*) es aquel que restringe la búsqueda a uno o más campos dados.

Una forma es indicar el campo en el que realizar la búsqueda siguiendo la siguiente sintaxis:

@campo:criterios

El siguiente ejemplo devuelve aquellos documentos que tengan el término **club** o **fanclub** en algún campo, limitando la búsqueda a aquellos cuyo campo **origin** sea **UK**:

```
127.0.0.1:6379> FT.SEARCH bands 'club|fanclub @origin:UK'
1) (integer) 2
2) "band:teenage-fanclub:en"
3) 1) "name"
   2) "Teenage Fanclub"
   3) "origin"
   4) "UK"
4) "band:bombay-bicycle-club:en"
5) 1) "name"
   2) "Bombay Bicycle Club"
   3) "origin"
   4) "UK"
127.0.0.1:6379>
```

Si deseamos realizar la búsqueda de uno o más términos en uno o más campos explícitamente, se puede utilizar el parámetro **INFIELDS**, cuya sintaxis es:

INFIELDS númeroDeCampos campo campo campo...

Ejemplo:

```
FT.SEARCH bands US INFIELDS 1 origin
```

Filtro numérico

Un **filtro numérico** (*numeric filter*) es aquel que restringe la búsqueda a los documentos que tienen un valor de campo numérico en un rango. La sintaxis de este parámetro es:

FILTER campo mínimo máximo

Al igual que con el comando **ZRANGE** de **Redis**, se puede usar **-inf** y **+inf** como valores mínimo y máximo. Para que un extremo límite sea inclusivo, hay que precederlo de un paréntesis.

Ejemplo:

```
FT.SEARCH bands @origin:UK FILTER year (1990 2000
```

Filtro de claves

Un **filtro de clave** (*key filter*) restringe la búsqueda a solo aquellos documentos cuya clave sea una de las indicadas. Estas claves se enumeran mediante el parámetro **INKEYS**:

INKEYS númeroDeClaves clave clave clave...

Resultado sin contenido

Un documento indexado puede tener su clave homóloga en la base de datos o puede no tenerla. Cuando es así, el motor de consulta devuelve el contenido del documento. Cuando no existe, por ejemplo, porque se añadió con el parámetro **NOSAVE**, devuelve **nil** como contenido. En cualquier caso, siempre devuelve las claves.

Si deseamos obtener únicamente las claves, se puede usar el parámetro **NOCONTENT**. En este caso, el resultado será una lista donde el primer elemento devuelve el número de documentos encontrados y el resto, las claves de los documentos.

Ejemplo:

```
127.0.0.1:6379> FT.SEARCH bands @origin:US NOCONTENT
1) (integer) 3
2) "band:the-national:en"
3) "band:black-rebel-motorcycle-club:en"
4) "band:hotel-lights:en"
127.0.0.1:6379>
```