

Artículo

[Alberto Fuentes](#) · 29 jul, 2019 Lectura de 4 min

Búsquedas en campos de texto libre de forma rápida

¡Hola a tod@s!

Hoy os traigo un artículo de Kyle Baxter sobre búsquedas de texto libre que vale la pena guardar como referencia :)

¿Os gustaría buscar de forma eficiente campos de texto libres almacenados en vuestra aplicación? ¿Lo habéis intentado alguna vez pero no habéis encontrado una manera que os ofrezca un buen rendimiento? Hay un truco especial que resuelve el problema :)

Como es habitual, si preferís la versión TL;DR (Demasiado largo, no lo he leído), podéis ir directamente al final del artículo, pero preferiríamos que leyeseis el artículo entero para evitar herir sentimientos.

Vamos a utilizar como ejemplo la clase `Sample.Company` que encontrareis en:

- IRIS - incluida en el paquete de ejemplos <https://github.com/intersystems/Samples-Data> que puedes descargar e instalar
- Caché / Ensemble - namespace SAMPLES

La clase `Sample.Company` tiene el campo `Mission` que es un texto generado al azar. Para el ejemplo, se han generado alrededor 256,246 empresas.

```
/// The company's mission statement.  
Property Mission As %String(MAXLEN = 200, POPSPEC = "Mission()");
```

Supongamos que queremos buscar en este campo de texto:

```
SELECT * FROM Sample.Company WHERE LIKE '%agile%'
```

Es una consulta bastante razonable, pero ¿cómo funciona? Al no tener un índice, claramente necesita leer cada entrada, de modo que obtenemos 256,277 referencias a globals de nuestras 7,854 filas devueltas. ¡Esto no está bien! Añadamos un índice a `Sample.Company` y veamos si podemos hacerlo mejor:

```
Index MissionIndex on Mission;
```

Construimos el índice y ejecutamos la misma consulta. ¿Qué obtenemos ahora? 279,088 referencias a globals.

Pero... ¡eso son más referencias a globals! ¿Acaso no es malo? ¡Los índices debían habernos ayudado!

Calmémonos. Además, cuando nos enfrentamos a un comportamiento contradictorio, es mejor tomarse un momento para pensar. ¿Cuál es el coste de leer un índice en comparación con leer la tabla de datos completa? Un índice será más pequeño, así que leer el índice `MissionIndex` completo será una operación menos costosa que hacer un escaneo completo de la tabla. Pero además del índice, necesitaremos leer algunas partes de la tabla para mostrar los resultados. En este caso, aunque haya más referencias a globals, es menos trabajo (asumiendo que todo está en disco). Por supuesto, podemos hablar mucho más sobre este comportamiento y la estructura de

bloques de bases de datos de Caché pero eso está fuera del ámbito de este artículo.

De acuerdo, así que nuestra primera modificación y el ahorro de nuestra búsqueda posiblemente funcionaron, pero ciertamente el resultado fue menor de lo que nos gustaría. Buscamos algo mucho más rápido, queremos menos referencias a globals y que sea sencillo de implementar ¿qué podemos hacer? la respuesta es un índice iFind .

Vamos a definir el índice de la siguiente forma:

```
Index MissioniFind on (Mission) as %iFind.Index.Basic;
```

Reconstruimos los índices `##class(Sample.Company).%BuildIndices()` y ahora necesitamos decirle a la consulta que utilice este nuevo índice:

```
SELECT * FROM Sample.Company WHERE %ID %FIND search_index(MissioniFind,'agile')
```

Ahora, esta consulta obtiene las mismas 7,854 filas, pero lo hace con 7,928 referencias a globals. ¡Son apenas más referencias a globals que filas! Lo cual es algo muy bueno.

¿Podemos combinar esto con otros índices? Sí, las combinaciones de índices funcionan muy bien con esta tecnología.

¿Existe alguna restricción? Pues sí. Necesita que el ID de la clase sea compatible con bitmaps (mapas de bits). Es decir, el ID de la tabla debe ser un número entero positivo.

¿Dónde puedo obtener más información? En la documentación, desde luego:

<https://irisdocs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GSQLSRCHtxtsrchindexing>

<https://irisdocs.intersystems.com/irislatest/csp/documatic/%25CSP.Documatic.cls?APP=1&LIBRARY=%25SYS&CLASSNAME=%25iFind.Index.Basic>

TL;DR (Demasiado largo, no lo he leído):

Para buscar en un campo de texto libre:

1. Añadir un índice iFind sobre el campo de texto libre:

```
Index <IndexName> on <FreeTextField> As %iFind.Index.Basic
```

2. Reconstruir índices de forma habitual, por ejemplo:

```
do ##class(Sample.Company).%BuildIndices()
```

3. Re-escribir la consulta SQL de la siguiente manera en la cláusula WHERE:

```
...WHERE ID %FIND search_index(<IndexName>,<Search Value>) AND ...
```

[#iFind](#) [#Indexación](#) [#Mejores prácticas](#) [#Modelo de datos de objetos](#) [#ObjectScript](#) [#SQL](#) [#Caché](#)

URL de

fuelle:<https://es.community.intersystems.com/post/b%C3%BAsuedas-en-campos-de-texto-libre-de-forma-r%C3%A1pida>