
Artículo

[Ricardo Paiva](#) · 23 sep, 2019 Lectura de 3 min

SQL y Funciones definidas por el usuario

¡Hola a tod@s!

¿Alguna vez se han encontrado con una query de bajo rendimiento a la cual le falta algún dato para obtener el rendimiento deseado? ¿Han tenido la necesidad de llamar a una función desde una sentencia SQL? Me refiero a funciones definidas por el usuario, no a las funciones ya disponibles (como \$LENGTH) o a las variables especiales (como \$HOROLOG). Si es así, tenemos solución para esta necesidad - las 'funciones extrínsecas' o 'funciones definidas por el usuario'. Se trata de un mecanismo potente con sus riesgos - hay que pensárselo bien a la hora de usarlo. Sin embargo, se trata de algo que puede aportar un incremento significativo en rendimiento a la hora de leer los datos. De igual manera se podría usar para cambiar alguno estado - pero este escenario no forma parte de este artículo.

Para plantearlo bien, pensemos en un caso de uso.

Tenemos la siguiente tabla:

```
Class User.Table1 Extends %Persistent [ SqlTableName = Table1 ]
{
Property Field1 As %Integer;
Property Field2 As %String;
Property Field3 As %String;
Property Field4 As %String;
Index Field1IDX On Field1;
Index Field3 On Field3 [ Type = bitmap ];
}
```

En esta tabla tenemos 30,000,000 de registros. Con esta carga, las búsquedas tienen que ser eficientes. Necesitamos sacar los datos del Field4 y no tenemos datos para hacer la búsqueda en base a los campos indexados. Hacemos la siguiente consulta:

```
select Field4 from sqluser.table1 where Field2 ='search data for Field2'
```

Se trata de una consulta demasiado ineficiente, a mí me tomó 25,502,965 referencias globales y unos 60.00189 segundos.

Tenemos un campo con un BitMap Index, lo que es una 'sugerencia' que debemos aprovechar. Sin embargo en este caso, el campo del índice bitmap tiene solamente 2 valores posibles. Si tuviésemos al valor de este campo podríamos hacer la siguiente consulta:

```
select Field4 from sqluser.table1 where Field2 ='search data for Field2' and Field3='
bitmap search value'
```

Algo hemos ganado con la utilización de este campo en nuestra consulta. A mí me tomó 15,000,475 referencias globales y 49.621 segundos.

Hasta aquí nada de nuevo. Ahora imaginemos que teníamos una función de usuario que nos permitía obtener el valor a usar en el Field1 a través de otro cualquier dato. Imaginemos que esta función (*.mac) fuera algo así:

```
TEST
Quit 0
GetValue(val)
Set tVal=$g(val)
Set tRet=$g(^zGlobal(tVal))
Quit tRet
```

Tenemos todo lo necesario para escribir una consulta así:

```
select Field4 from sqluser.table1 where Field3='bitmap search value' and Field1=$$GetValue^TEST("some other key value")
```

Esta última consulta a mí me tomó 0.006 segundos y 13 referencias globales. Obviamente en esta consulta se incluye el gasto (tiempo y recursos) de la función GetValue en la rutina TEST. Para comprobarlo habrá que ejecutar la consulta:

```
select Field4 from sqluser.table1 where Field3='bitmap search value' and Field1=IDKey
```

Esta consulta me tomó 0.002 segundos y 12 referencias globales. Queda claro que la utilización de la función de usuario resulta beneficiosa en este caso. Es una opción en casos puntuales pero no es SQL estándar. Por lo tanto su uso habrá de ser ponderado. Para que todo esto funcione, habrá que activar el uso de 'funciones extrínsecas'. Para eso habrá que ejecutar el comando:

```
USER>Do $SYSTEM.SQL.SetAllowExtrinsicFunctions()
```

[#Code Snippet](#) [#Globals](#) [#InterSystems IRIS](#)

URL de fuente: <https://es.community.intersystems.com/post/sql-y-funciones-definidas-por-el-usuario>