

Artículo

[Muhammad Waseem](#) · 19 jul, 2022 Lectura de 3 min

Índices únicos y valores nulos en InterSystems IRIS

Recientemente surgió un patrón interesante en torno a los índices únicos (en una discusión interna re: [isc.rest](#)) y me gustaría destacarlo para la Comunidad.

Como caso de uso motivador: supón que tienes una clase que representa un árbol, donde cada nodo también tiene un nombre, y queremos que los nodos sean únicos por nombre y nodo principal. Queremos que cada nodo raíz también tenga un nombre único. Una implementación natural sería:

```
Class DC.Demo.Node Extends %Persistent
{
    Property Parent As DC.Demo.Node;
    Property Name As %String [ Required ];
    Index ParentAndName On (Parent, Name) [ Unique ];
    Storage Default
    {
        <Data name="NodeDefaultData">
        <Value name="1">
        <Value>%%CLASSNAME</Value>
        </Value>
        <Value name="2">
        <Value>Parent</Value>
        </Value>
        <Value name="3">
        <Value>Name</Value>
        </Value>
        </Data>
        <DataLocation>^DC.Demo.NodeD</DataLocation>
        <DefaultData>NodeDefaultData</DefaultData>
        <IdLocation>^DC.Demo.NodeD</IdLocation>
        <IndexLocation>^DC.Demo.NodeI</IndexLocation>
        <StreamLocation>^DC.Demo.NodeS</StreamLocation>
        <Type>%Storage.Persistent</Type>
    }
}
```

¡Y aquí estamos!

Pero hay un problema: tal y como está, esta implementación permite que varios nodos raíz tengan el mismo nombre. ¿Por qué? Porque Parent no es (y no debería ser) una propiedad requerida, e IRIS no trata "nulo" como un valor distinto en índices únicos. Algunas bases de datos (por ejemplo, SQL Server) lo hacen, pero el estándar SQL dice que no es correcto [cita requerida; vi esto en StackOverflow en alguna parte, pero eso realmente no cuenta; echa un vistazo al comentario de [@Dan Pasco](#) en esta publicación sobre esto y la distinción entre índices y restricciones].

La forma de evitar esto es definir una propiedad calculada que se establezca en un valor no nulo si la propiedad a la que se hace referencia es nula y luego poner el índice único en esa propiedad. Por ejemplo:

```
Property Parent As DC.Demo.Node;  
Property Name As %String [ Required ];  
Property ParentOrNUL As %String  
  [ Calculated, Required, SqlComputeCode =  
{Set {*} = $Case({Parent}, "":$c(0),:{Parent})}, SqlComputed ];  
Index ParentAndName On (ParentOrNUL, Name) [ Unique ];
```

Esto también permite pasar \$c(0) a ParentAndNameOpen/Delete/Exists para identificar un nodo raíz únicamente por padre (no hay uno) y nombre.

Como ejemplo motivador en el que este comportamiento es muy útil, consulta <https://github.com/intersystems/isc-rest/blob/main/cls/pkg/isc/rest/res...>. Muchas filas pueden tener el mismo conjunto de valores para dos campos (DispatchOrResourceClass y ResourceName), pero queremos que, como máximo, uno de ellos se trate como el "predeterminado", y un índice único funciona perfectamente para hacer cumplir esto si decimos que el indicador "predeterminado" se puede establecer en 1 o nulo y después poner un índice único en él y los otros dos campos.

[#Indexación](#) [#SQL](#) [#Caché](#) [#InterSystems IRIS](#)

URL de
fuente: <https://es.community.intersystems.com/post/%C3%ADndices-%C3%BAnicos-y-valores-nulos-en-intersystems-iris>