Artículo
Ricardo Paiva · 3 dic, 2021 Lectura de 2 min

Tablas y clases semi-persistentes

Si defines una tabla/clase persistente, el compilador de clases genera una definición de almacenamiento adecuada. Otra opción es definir un mapeo SQL para un almacenamiento global que ya existe. Esto ya se explicó estupendamente en otra serie de artículos: El arte del mapeo de globales para Clases 1 de 3

Después de definir el diagrama de almacenamiento, el compilador de la clase lo podrá ampliar, pero los parámetros fundamentales de almacenamiento no cambiarán.
Esto no significa que no puedas cambiarlo manualmente.

Mi artículo El mapa de bits adoptado incluye un caso como este. Y en combinación con algunas Condiciones WHERE estáticas, como se describió anteriormente, permite que esté disponible también para SQL.

La definición habitual de tus *globals* de almacenamiento se parecerá a este ejemplo:

```
 <DataLocation>^User.PersonD</DataLocation>
<IdLocation>^User.PersonD</IdLocation>
<IndexLocation>^User.PersonI</IndexLocation>
<StreamLocation>^User.PersonI</StreamLocation>
```

Ahora cambiaremos esta definición por una dinámica

```
 <DataLocation>@%MyData</DataLocation>
<IdLocation>@%MyId</IdLocation>
<IndexLocation>@%MyIndex</IndexLocation>
<StreamLocation>@%MyStream</StreamLocation>
```

y añadimos algo de comodidad

```
 Parameter MANAGEDEXTENT As INTEGER = 0;

ClassMethod SetStorage(ref As %String) As %Integer [ SqlName = SetStorage, SqlProc ]
{
  set %MyData=ref_"D"
  , %MyId=%MyData
  , %MyIndex=ref_"I"
  , %MyStream=ref_"S"
  quit $$$OK
}
```

Para acceder a los objetos, dirigimos nuestro almacenamiento y lo llenamos

```
 write ##class(Person).SetStorage("^mtemp.Person")
write ##class(Person).Populate(5)
```

y en SQL:

```
 SELECT  * from Person where SetStorage('^mtemp.Person')=1
```

Funciona con PPG (**^||Person**)
mediante el *namespace* (**^|"USER"|Person**)
también corresponde al subíndice que se utilizó en El mapa de bits adoptado con otra variante de **ClassMethod SetStorage()**

Para acceso a objetos (sin SQL), funciona incluso para variables locales.

No es el uso deseado, pero demuestra la flexibilidad de esta función.

Pero ten cuidado. **NO** funcionará con *sharding*. Pero eso no es una sorpresa.

#Code Snippet #Globals #Mapeo #Modelo de datos de objetos #ObjectScript #SQL #Caché #Ensemble #InterSystems IRIS

URL de fuente:https://es.community.intersystems.com/post/tablas-y-clases-semi-persistentes