

# El arte de mapear Globals para Clases (2 de 3)

Artículo

[Estevan Martinez](#) · Jun 19, 2019



Lectura de 8 min

## El arte de mapear Globals para Clases (2 de 3)

¿Alguno de ustedes desea traer de vuelta una antigua aplicación en MUMPS? Siga estos pasos para mapear sus globales ya establecidos para clases y mostrar todos esos hermosos datos en Objects y SQL

En este ejemplo se estudiarán 4 o 5 cosas diferentes más allá de lo que se cubrió en la [Parte 1](#)

Todo lo que falta por hacer después de esto, es el ejemplo del mapeo de padres e hijos, entonces irá por buen camino.

**Descargo de responsabilidad:** Si sus variables globales todavía no tienen ni pies ni cabeza después de revisar estos artículos, póngase en contacto con el Centro de Soporte Internacional (WRC) y le ayudaremos a resolver su incidencia: [Support@InterSystems.com](mailto:Support@InterSystems.com)

Pasos para mapear una variable global para una clase:

1. Identifique algún patrón que se repita en los datos globales.
2. Identifique que la hace una clave única.
3. Identifique las propiedades y sus categorías.
4. Defina las propiedades en la clase (no olvide las propiedades de los subíndices de las variables).
5. Defina el índice IdKey.
6. Establezca la definición de almacenamiento:
  - a. Establezca los subíndices, e incluya la IdKey.
  - b. Establezca una sección para los datos.
  - c. Ignore la sección "Row ID (Identificación de la fila)". El 99% de las veces el valor que desea es el predeterminado, así que deje que el sistema lo llene por usted.
7. Compile y pruebe su clase / tabla.

Digamos que tiene 2 globales que se parecen a esto:

```
^mapping("Less Simple",1,1)="Bannon,Brendan^Father"  
^mapping("Less Simple",1,1,"Activity")="Rock Climbing"  
^mapping("Less Simple",1,2)="Bannon,Sharon^Mother"  
^mapping("Less Simple",1,2,"Activity")="Yoga"  
^mapping("Less Simple",1,3)="Bannon,Kaitlin^Daughter"  
^mapping("Less Simple",1,3,"Activity")="Lighting Design"  
^mapping("Less Simple",1,4)="Bannon,Melissa^Daughter"  
^mapping("Less Simple",1,4,"Activity")="Marching Band"  
^mapping("Less Simple",1,5)="Bannon,Robin^Daughter"  
^mapping("Less Simple",1,5,"Activity")="reading"  
^mapping("Less Simple",1,6)="Bannon,Kieran^Son"  
^mapping("Less Simple",1,6,"Activity")="Marching Band"  
  
^index("Less Simple","FName","BRENDAN",1,1)=" "  
^index("Less Simple","FName","KAITLIN",1,3)=" "
```

```
^index("Less Simple", "FName", "KIERAN", 1, 6) = ""
^index("Less Simple", "FName", "MELISSA", 1, 4) = ""
^index("Less Simple", "FName", "ROBIN", 1, 5) = ""
^index("Less Simple", "FName", "SHARON", 1, 2) = ""
```

### Paso 1:

Parece que esta vez los datos repetidos se reparten en dos nodos globales en lugar de uno.

### Paso 2:

Es difícil asegurarlo con solo esta parte de los datos. Ya sea el primero o el primero y el segundo subíndices son constantes. Para este ejemplo asumiremos que el segundo subíndice es una variable, `FamilyId`, y el tercer subíndice es `PersonId`.

### Paso 3:

Así que tenemos los dos subíndices variables `FamilyId` y `PersonId`, y luego tenemos el nombre, la relación y la actividad. Miremos de nuevo el índice global para buscar alguna pista, voy a mapear el nombre de las dos propiedades, `FirstName` y `LastName`. Ahora tenemos un total de 6 propiedades que necesitamos definir.

### Paso 4:

Hay un par de cosas que son nuevas aquí. Todas las propiedades tienen un SQL Field Name, mientras que `FirstName` tiene una compilación para la función `UPPER`. Definir los nombres de los campos SQL, hará que tengamos más claro el procedimiento sobre cuándo hablamos de una propiedad y cuando de un campo SQL. Hablaremos de la compilación cuando veamos cómo mapear el índice.

```
Property FamilyId As %Integer [ SqlFieldName = Family_Id ];
Property PersonId As %Integer[ SqlFieldName = Person_Id ];
Property FirstName As %String(COLLATION = "UPPER")[ SqlFieldName = First_Name
];
Property LastName As %String[ SqlFieldName = Last_Name ];
Property Relation As %String[ SqlFieldName = Relation ];
Property Activity As %String[ SqlFieldName = Activity ];
```

### Paso 5:

En las variables hay dos subíndices, lo cual significa que la `IdKey` se basará en dos propiedades:

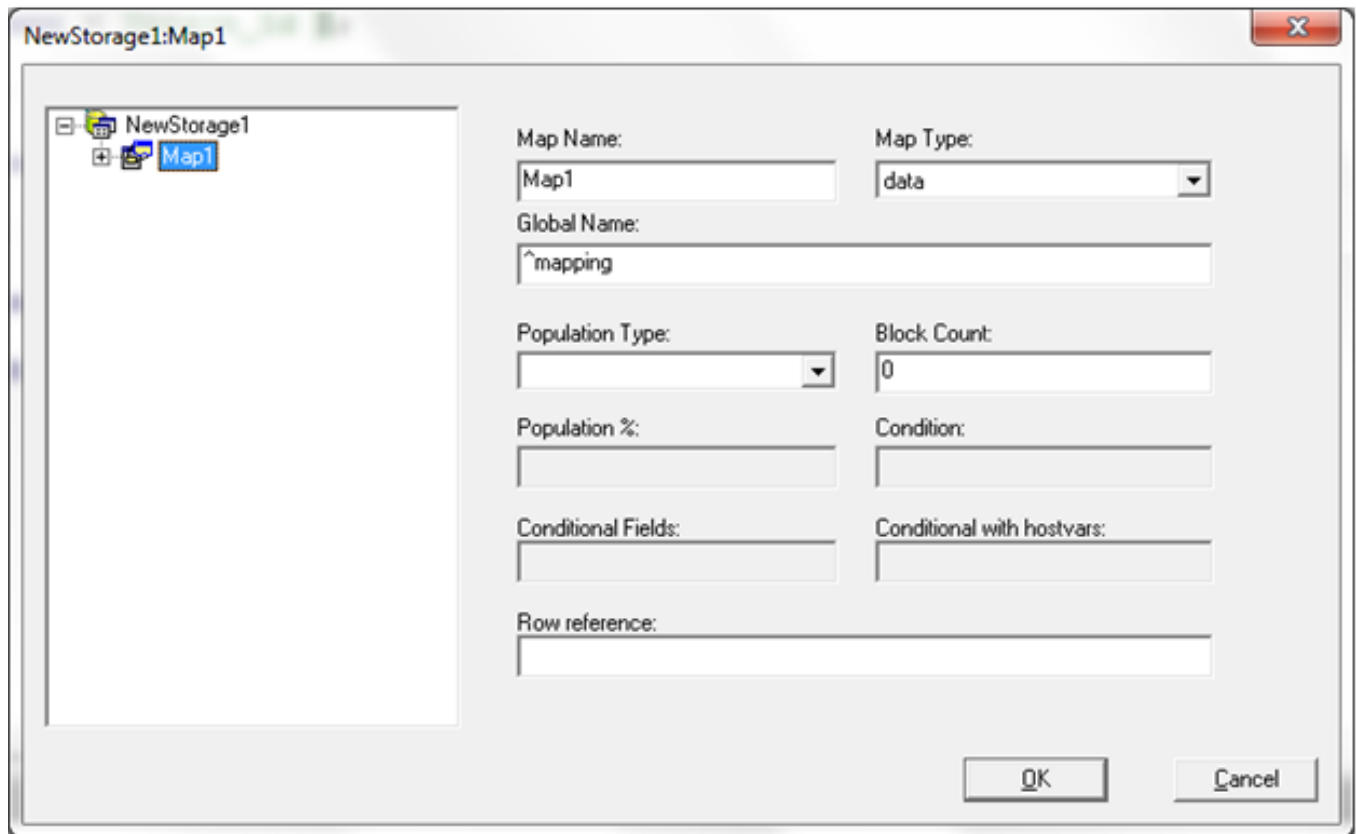
```
Index Master On (FamilyId, PersonId)[IdKey];
```

Nuevamente tenemos 1 índice definido sobre 1 propiedad, `FirstName`.

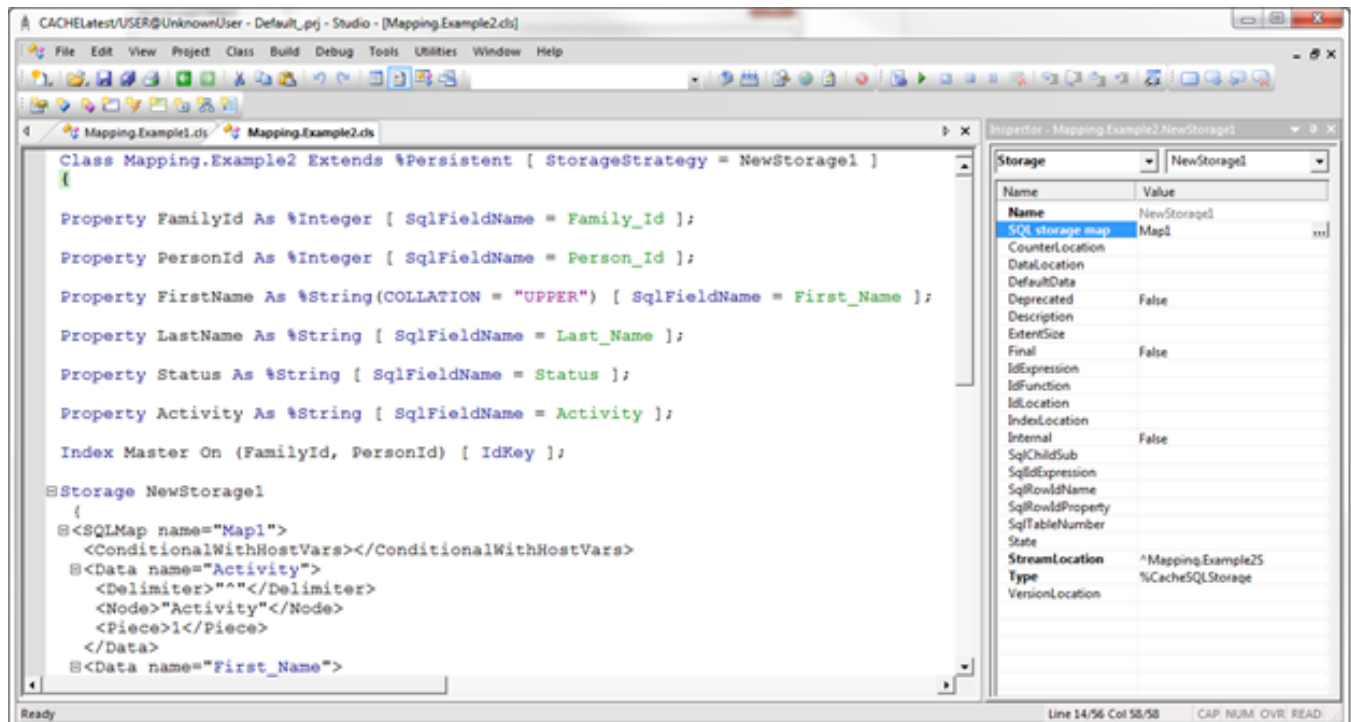
```
Index FNameIndex On FirstName;
```

### Paso 6:

Lo primero que debemos hacer es crear la definición de almacenamiento. Haga clic en el icono "Storage" o utilice el Inspector, Storage, y luego haga clic con el botón derecho del ratón en "New Storage". Esta vez dejé el valor predeterminado para el campo "Map Name", y solamente llene la casilla "Global Name" con `^mapping`.



Algo que no le mostré en el primer ejemplo es cómo regresar al asistente después de haber creado una definición de almacenamiento. Desde "Inspector" seleccione "Storage", luego elija el nombre del almacenamiento (en este caso NewStorage1), haga clic sobre "SQL Storage Map" y haga clic nuevamente en el cuadro que aparece en el lado derecho para abrir el asistente.

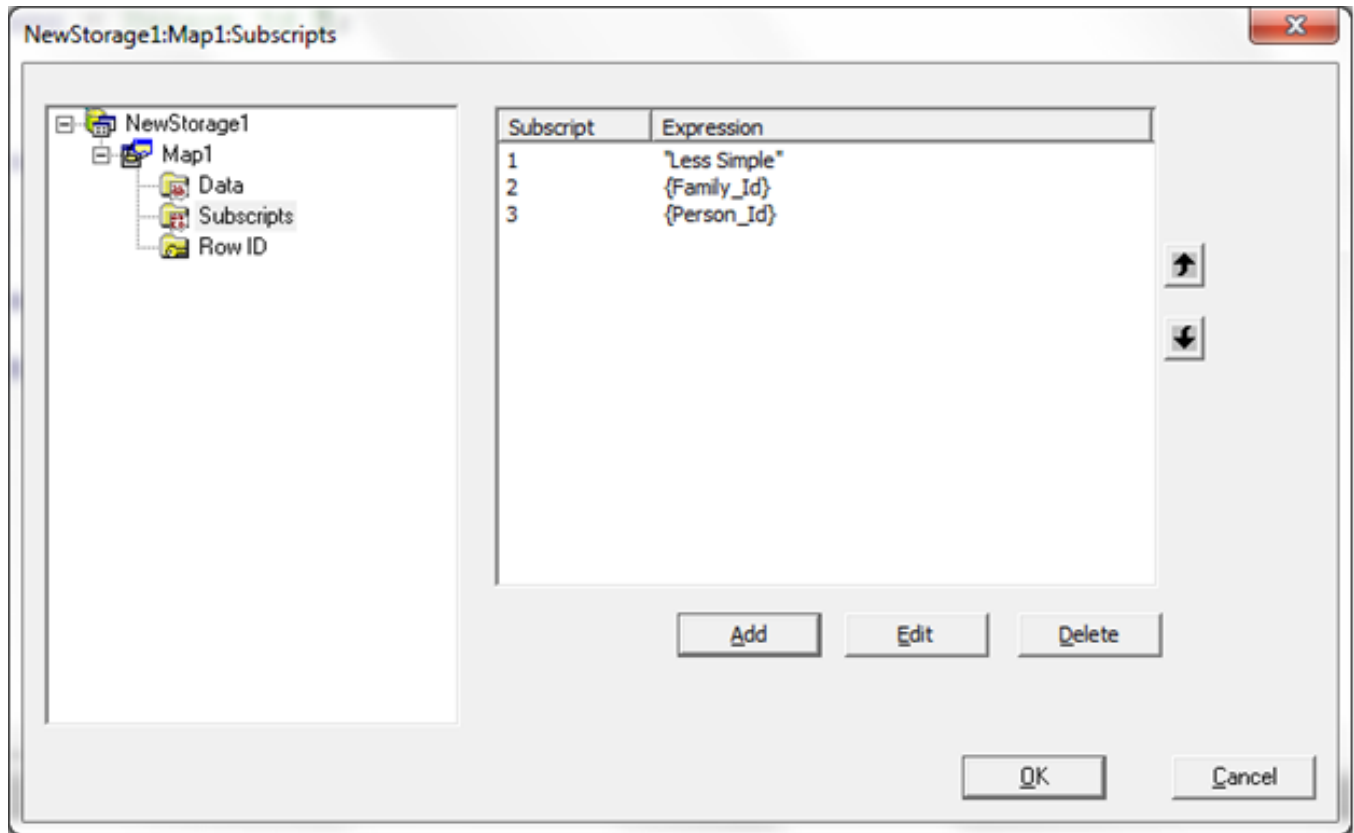


**Paso 6a:**

Comencemos de nuevo con la sección "Subscripts (Subíndices)". Primero necesitaremos 3 niveles de subíndice para definir todo, incluso la IdKey: 1 constante y 2 campos. Los campos que aparecen en esta ventana deben

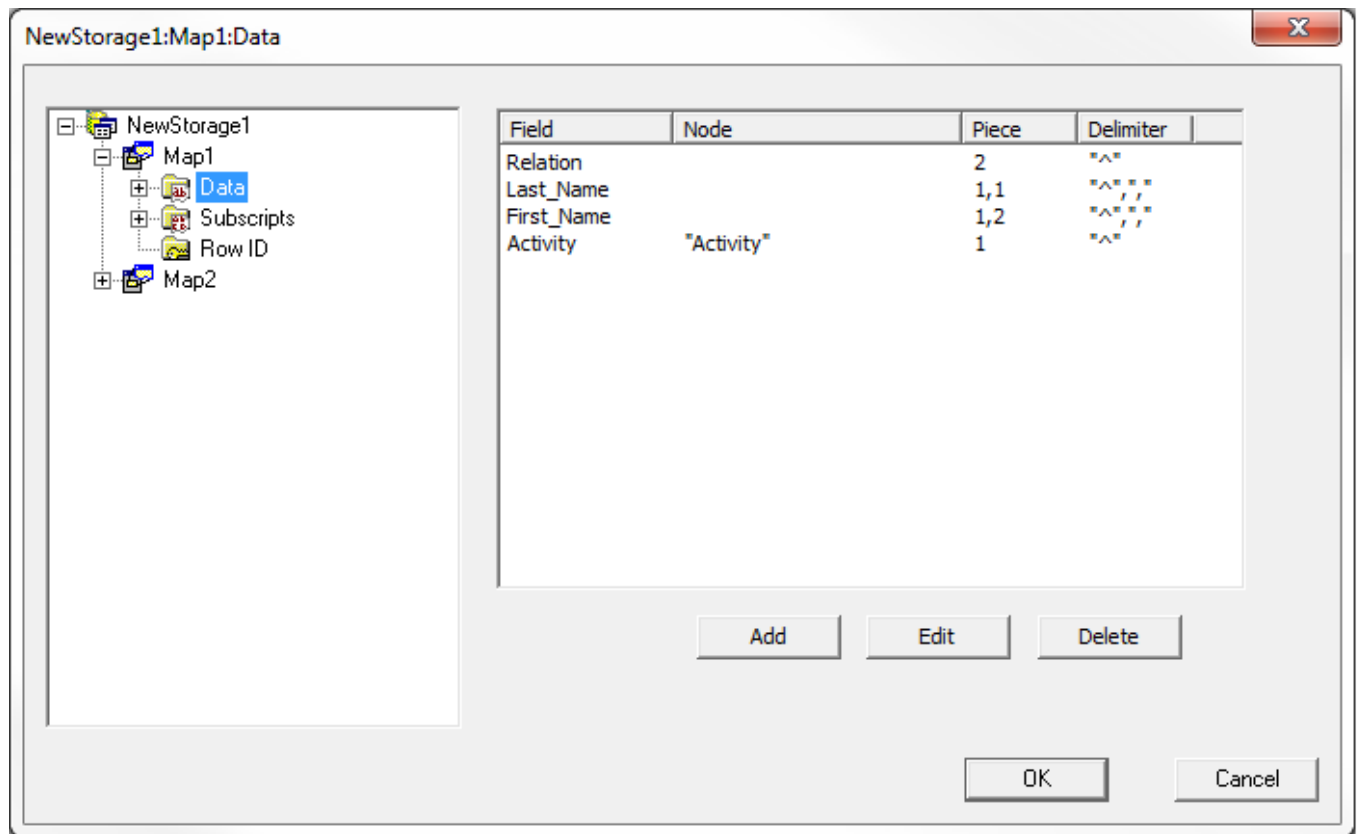
coincidir con las propiedades definidas para el índice IdKey.

Nota: En la definición de almacenamiento siempre nos referiremos al nombre del campo SQL, `Family_Id`, no al nombre de la propiedad, `FamilyId`.



### Paso 6b:

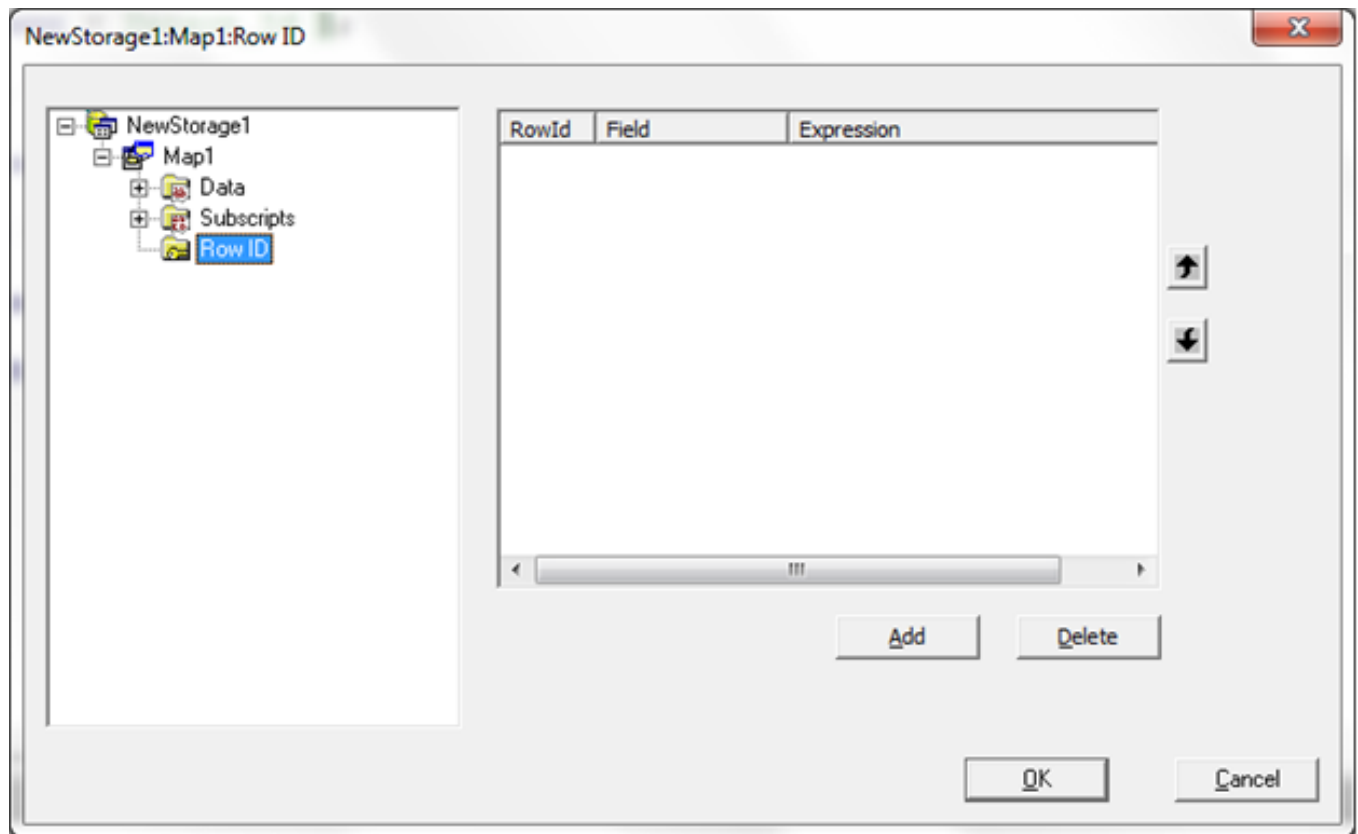
Para la sección "Data" tenemos 3 campos almacenados en 1 nodo global, y luego tenemos 1 campo almacenado en un nivel de subíndice inferior ("Activity"). Para obtener el nombre y el apellido como dos campos diferentes necesitaremos usar múltiples piezas y los delimitadores correspondientes para un `$PIECE()` anidado. En ObjectScript esto se vería como: `set FirstName=$PIECE($PIECE(^mapping("Less Simple",1,1),"^",1),",",",",2)`. También necesitamos hacer una lista en el asistente, con todos los delimitadores y las piezas comenzando desde la parte más interna de `$PIECE()` y trabajando hacia afuera. El subíndice inferior Activity está en un nodo global diferente, por lo que añadimos la constante a Node. Piece y Delimiter están por defecto en 1 al igual "^", así que los dejé allí.



### Paso 6c:

Sigue sin haber nada que ver aquí.

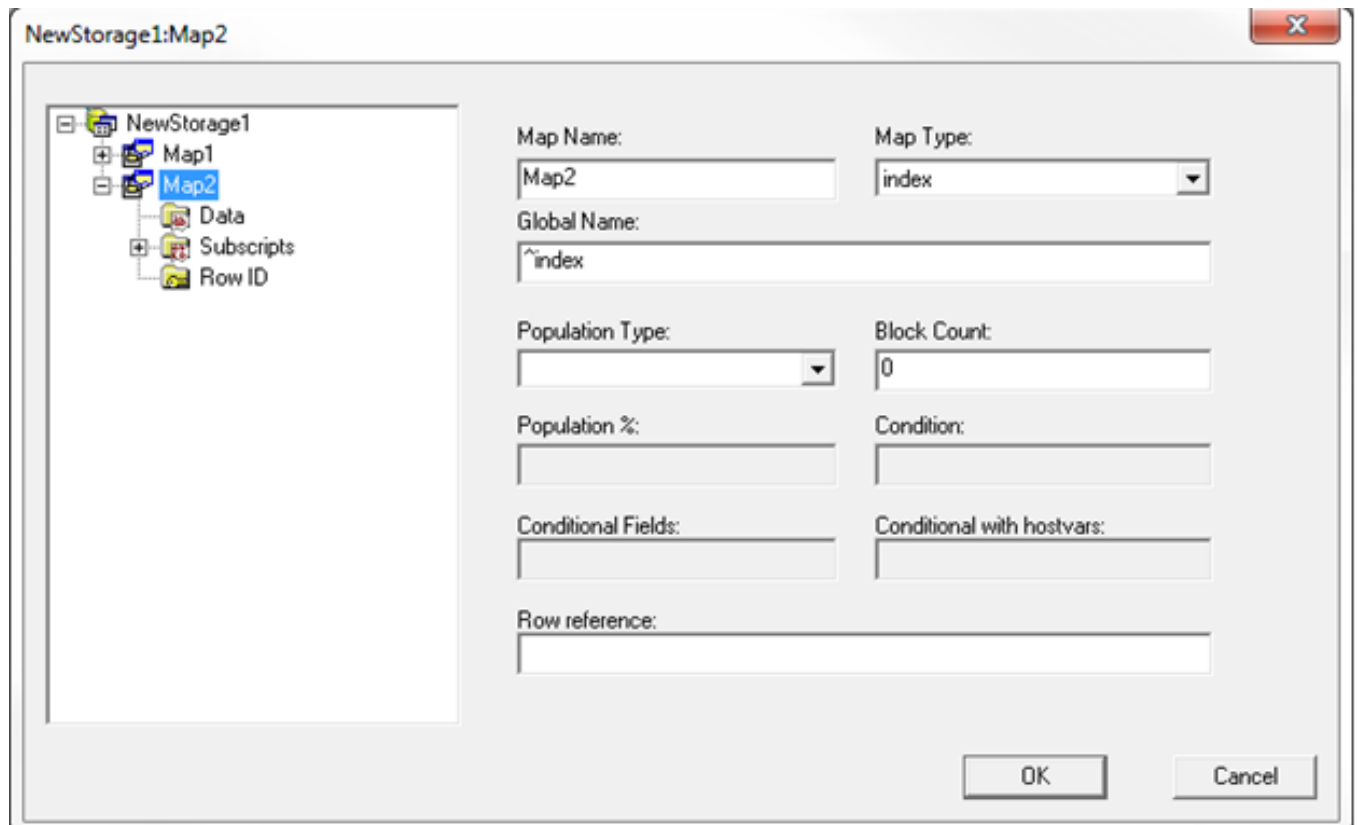
Muy bien, sé que tiene muchas ganas de ver un ejemplo sobre cuándo necesita definir la sección "Row ID". Cuando publique el archivo zip con todos los ejemplos, habrá uno llamado `Mapping.RowIdSpec` en el que se muestra un caso donde es necesario hacer esto.



Ahora repetiremos el Paso 6 para crear el índice del mapa.

**Paso 6**

Cree Map2 y establezca el nombre de la variable global en ^index.



### Paso 6a

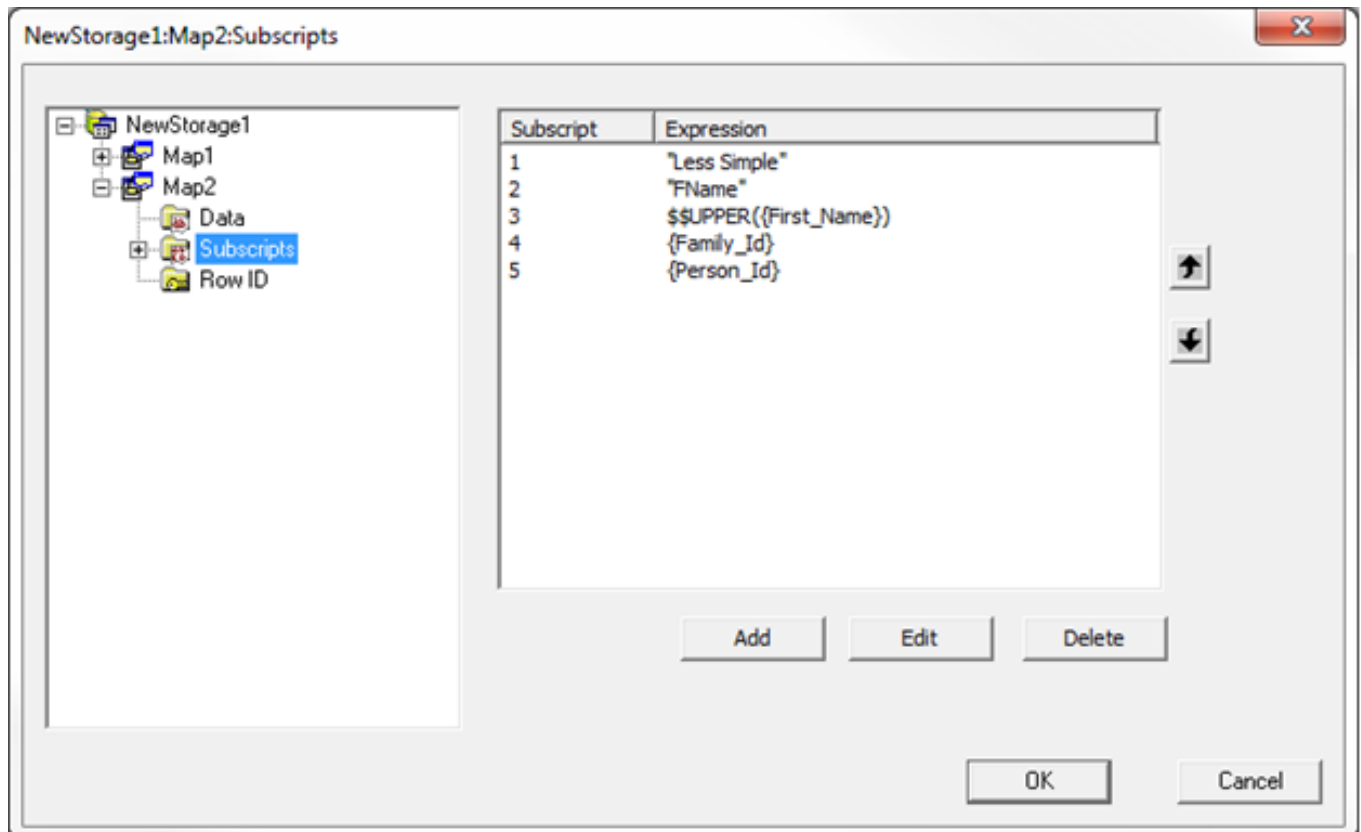
El index global tiene 5 subíndices: 2 constantes, 2 campos IdKey, y el campo indexado, `First_Name`. Si miramos el `^mapping` global y el `^index` global, los valores de `First_Name` son diferentes. En el `^mapping` global los nombres tienen mayúsculas y minúsculas, mientras que en el `^index` global los nombres son todos en mayúsculas. Caché incorpora varias funciones de compilación, las cuales puede utilizar para transformar cadenas de datos. Consulte el siguiente enlace, para obtener más información sobre las diferentes funciones que están disponibles para usted:

[http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GSQL\\_basics#GSQL\\_basics\\_collation](http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GSQL_basics#GSQL_basics_collation)

Para Caché SQL Storage seguramente necesitará consultar las clases anteriores sobre los tipos de compilación disponibles:

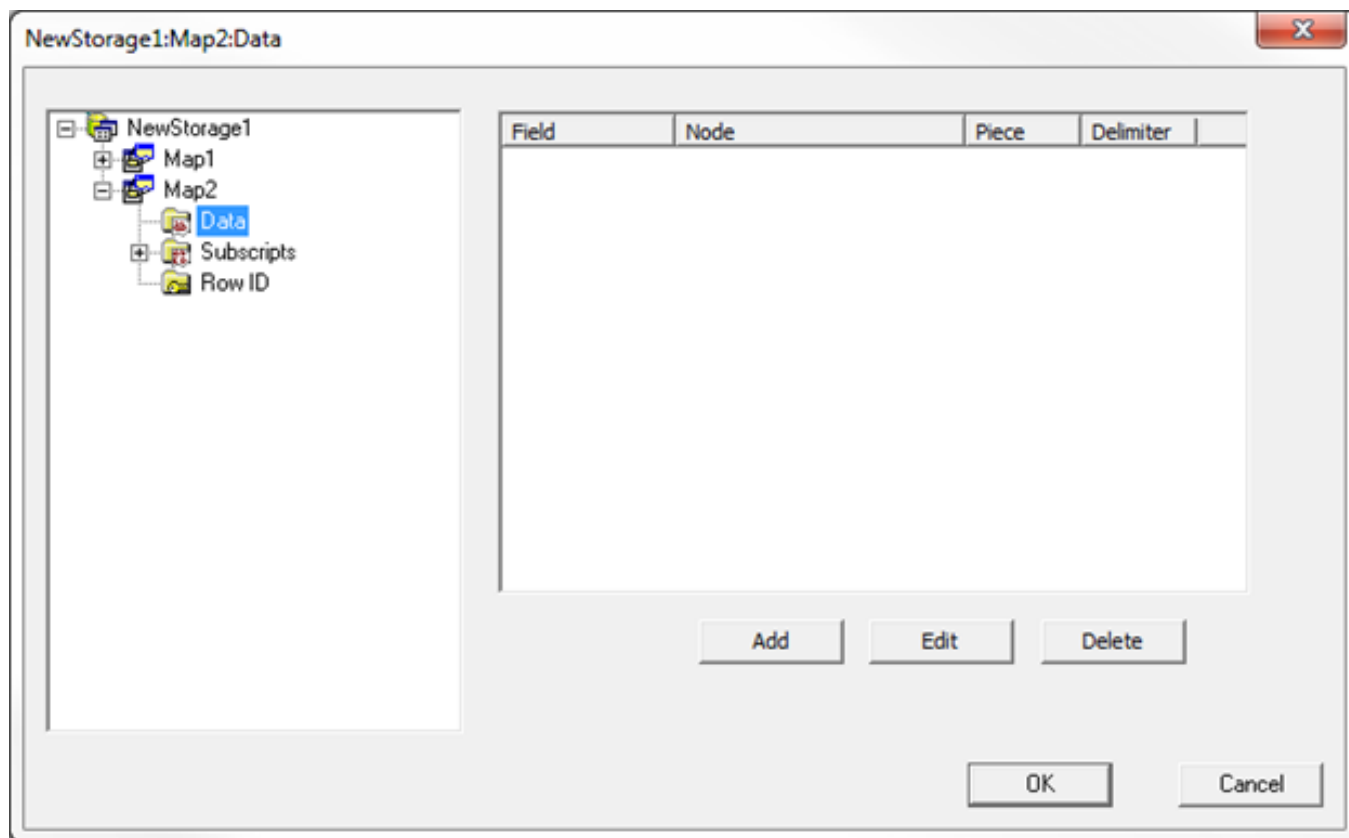
[http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GSQL\\_basics#GSQL\\_basics\\_collation\\_legacy](http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GSQL_basics#GSQL_basics_collation_legacy)

Es muy importante que la función de compilación que se utilice aquí, coincida con la función de compilación cuando se defina la propiedad. La causa más frecuente de que el optimizador de consultas no utilice el índice del mapa es porque éste no coincide. Recuerde que `%String` tiene una compilación predeterminada de `SQLUPPER`, y que no proporcionar una función durante el mapeo sería igual que la compilación en `EXACT`.



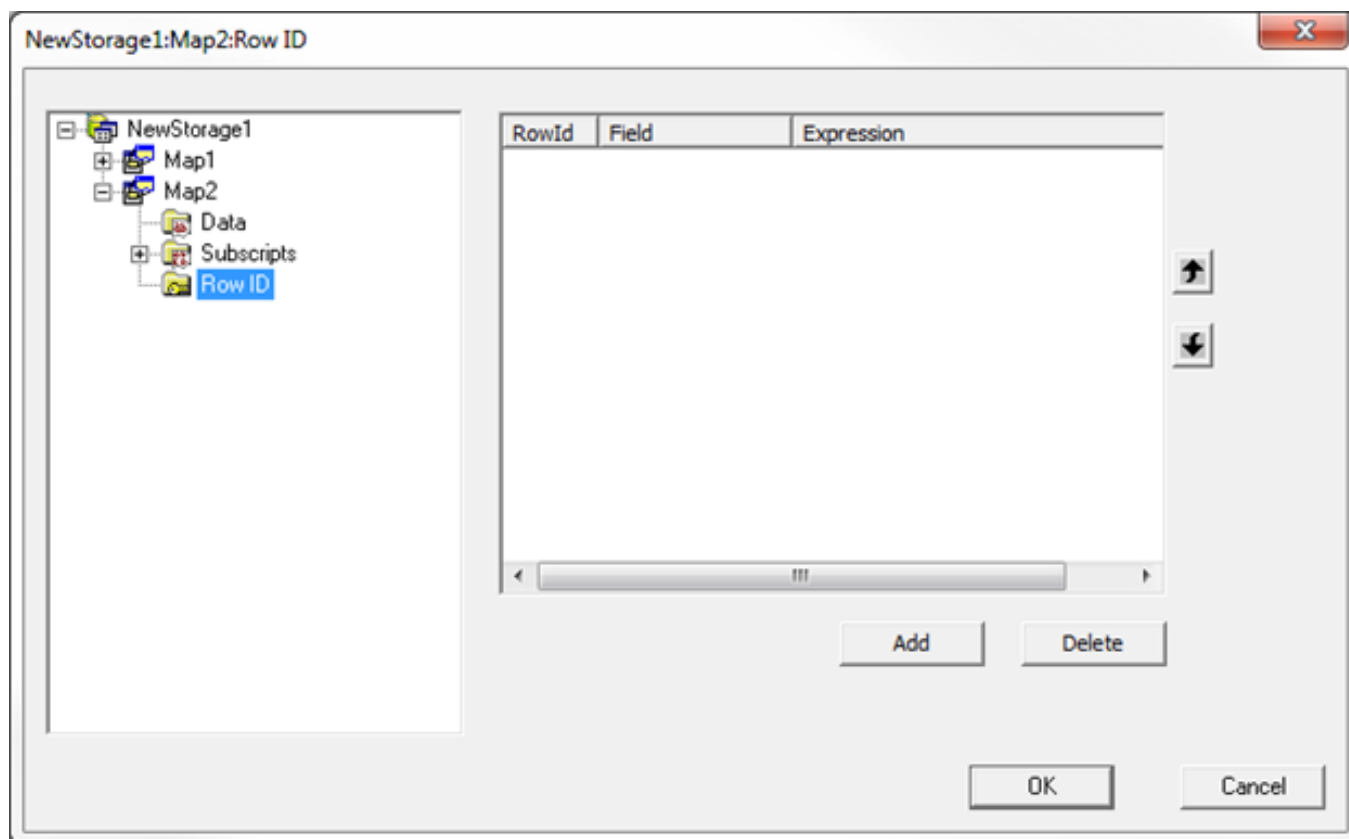
### Paso 6b

El `^index` global no contiene datos, entonces no hay nada que definir aquí.



**Paso 6c**

Como ya debe saber, aquí tampoco hay algo.



**Paso 7:**



## El arte de mapear Globals para Clases (2 de 3)

Published on InterSystems Developer Community (<https://community.intersystems.com>)

---

```
Compilation started on 08/22/2016 15:42:16 with qualifiers 'fck
/checkuptodate=expandedonly'
Compiling class Mapping.Example2
Compiling table Mapping.Example2
Compiling routine Mapping.Example2.1
Compilation finished successfully in 0.144s.
```

```
SELECT Family_ID, Person_ID, First_Name, Last_Name, Relation, Activity
FROM Mapping.Example2
```

Family_Id	Person_Id	First_Name	Last_Name	Relation	Activity
1 Climbing	1	Brendan	Bannon	Father	Rock
1	2	Sharon	Bannon	Mother	Yoga
1 Design	3	Kaitlin	Bannon	Daughter	Lighting
1 Band	4	Melissa	Bannon	Daughter	Marching
1	5	Robin	Bannon	Daughter	Reading
1 Band	6	Kieran	Bannon	Son	Marching

Esta vez no les voy a molestar con la definición de almacenamiento. Si quieren verla, pueden descargar el xml y consultarla allí: [mapping.example2.zip](#)

[#Globals](#) [#Mapeo](#) [#SQL](#) [#Caché](#)

00 1 0 0 95

Mensajes relacionados

- [El arte de mapear Globals para Clases \(2 de 3\)](#)
- [El arte de mapear Globals para Clases \(3 de 3\)](#)
- [El arte de mapear Globals para Clases \(4 de 3\)](#)

[Show all](#)

Log in or sign up to continue  
Añade la respuesta

**URL de fuente:** <https://es.community.intersystems.com/post/el-arte-de-mapear-globals-para-clases-2-de-3>