Cargando tipos de datos compuestos

Objeto GeneXus: Data Provider

GeneXus™

En videos anteriores hemos visto el concepto de tipo de dato estructurado, la posibilidad de definirlos simples o como colecciones, y algunos ejemplos de carga manual, si bien hemos mencionado que existe otra manera de más de alto nivel para realizar la carga.

Nuevo requerimiento: Ranking de países

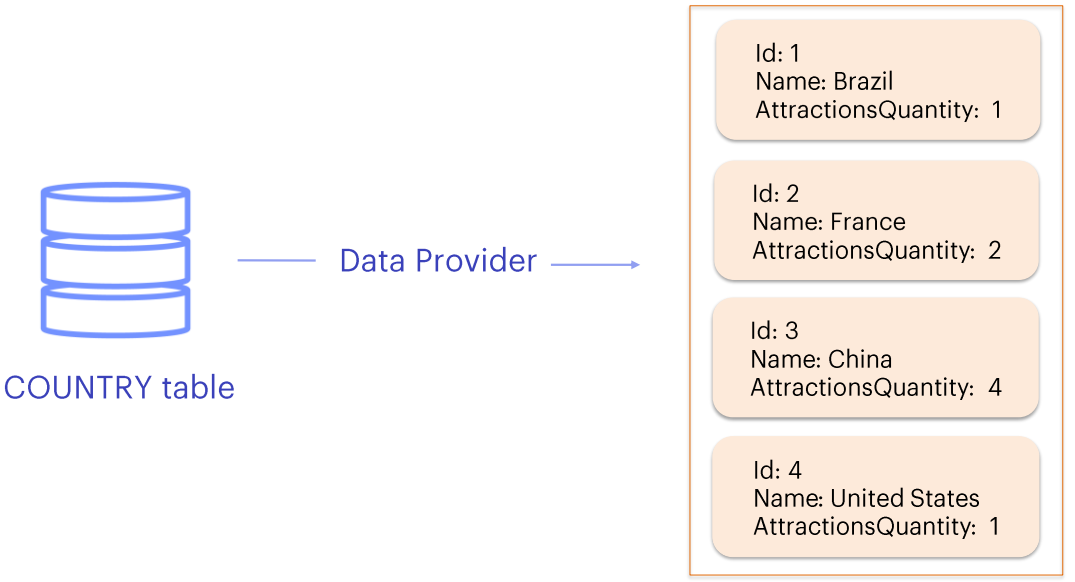
Ranking

Country name	Attractions quantity
France	3
United States	2
Egypt	1
Brazil	1
China	1
Uruguay	0

Si recordamos, tenemos pendiente resolver un requerimiento solicitado por la Agencia de Viajes que consiste en implementar un ranking de países según la cantidad de atracciones turísticas que ofrece.

Es decir que debemos mostrar todos los países, ordenados de mayor a menor, por dicha cantidad.

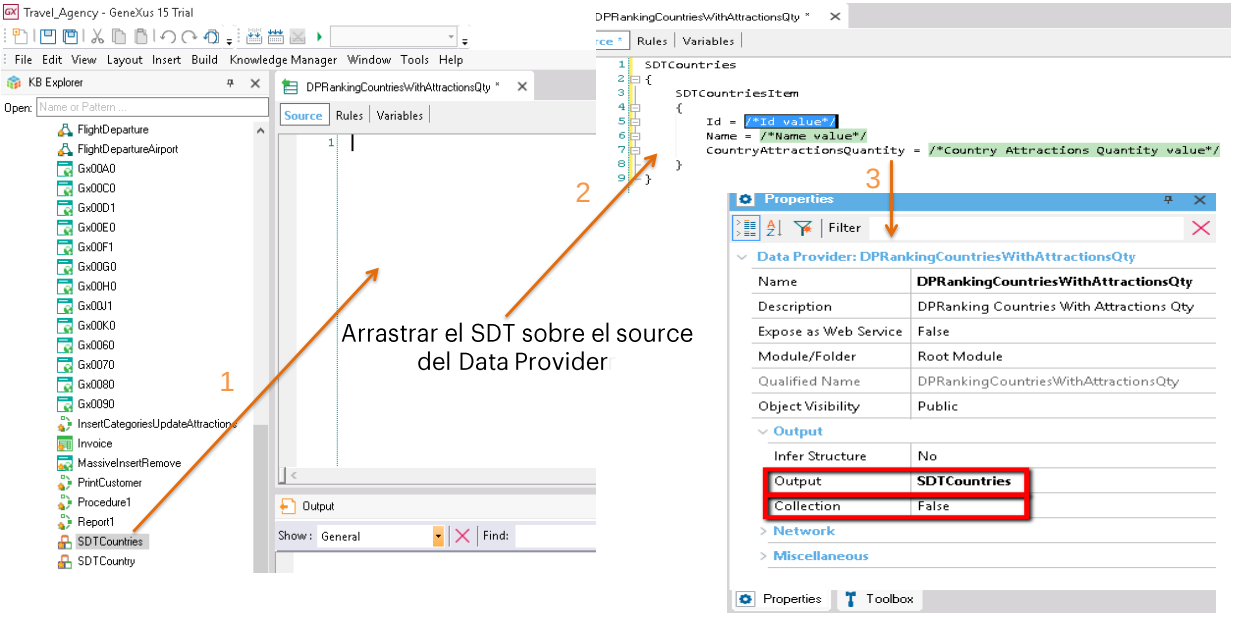
Colección de países



Para cargar los datos de la colección vamos a utilizar un objeto GeneXus de tipo **Data Provider**.

Este objeto nos permite cargar una estructura de datos, por ejemplo a partir de información obtenida de la base de datos, y nos devuelve dicha estructura ya cargada.

Nuevo requerimiento: Ranking de países



Creamos un objeto Data Provider en GeneXus y lo nombramos: DPRankingCountriesWithAttractionsQty.

GeneXus nos posiciona en la sección Source del Data Provider. Aquí es donde vamos a declarar cómo queremos que se carguen los datos en la colección que queremos devolver. Observemos qué fácil resulta declarar la carga: Vamos a la ventana del KB Explorer, ubicamos al tipo de datos estructurado SDTCountries y lo arrastramos hacia el source.

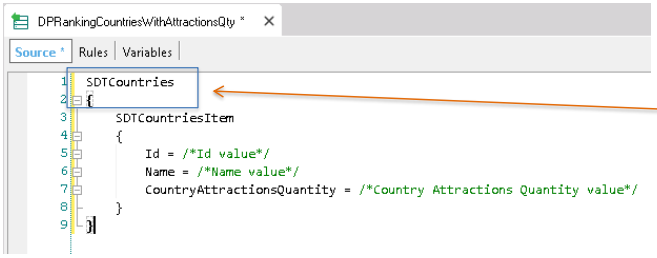
Por realizar el arrastre, GeneXus escribe automáticamente varias líneas de texto.

Si abrimos las propiedades del DataProvider, observamos que GeneXus asignó el nombre de la colección SDTCountries a la propiedad Output. **Esto significa que el DataProvider devolverá una colección del tipo de datos estructurado SDTCountries, cargada con datos.**

Como el SDTCountries ya es una colección, no es necesario configurar la propiedad Collection con valor True.

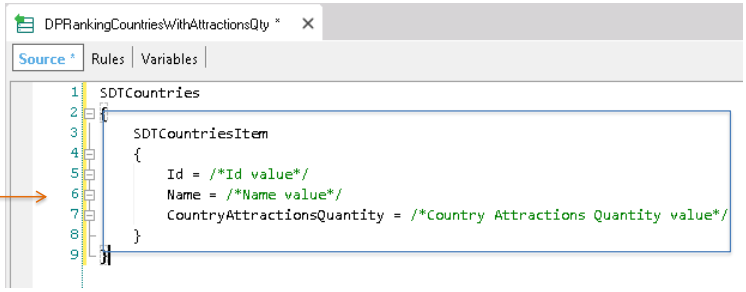
Es importante mencionar que en caso de indicar esta propiedad Collection con valor True, el DataProvider devolverá una colección del SDT que se haya indicado en la propiedad Output, sin importar lo compleja que pueda ser su estructura.

Nuevo requerimiento: Ranking de países



Nombre del tipo de datos estructurado.

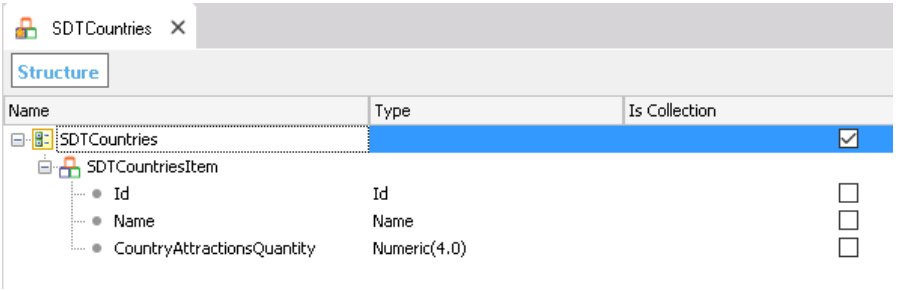
Subestructura del item de la colección.



Estudiemos ahora qué fue lo que escribió GeneXus en el source.

Reconocemos el nombre del tipo de datos estructurado SDTCountries que es una colección. Y después entre llaves está la subestructura del ítem de la colección.

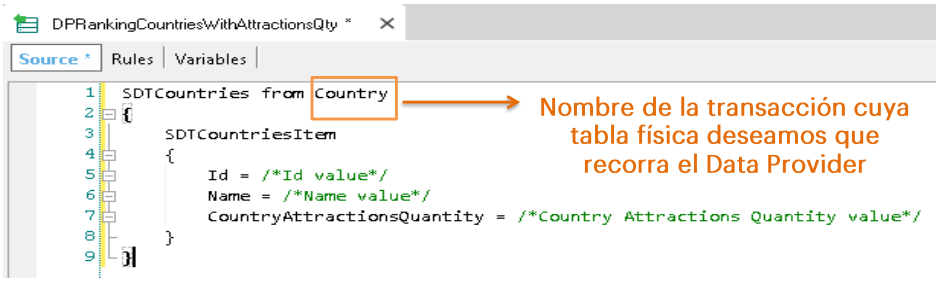
Nuevo requerimiento: Ranking de países



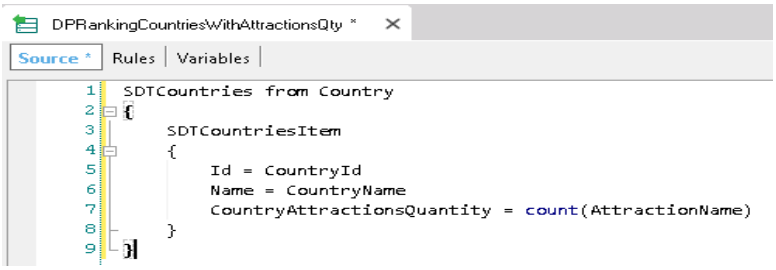
Comparemos esto con la estructura del SDT:

Vemos que GeneXus representó en forma de texto la estructura del SDTCountries. Y nos dejó prontos los miembros Id, Name y CountryAttractionsQuantity de la subestructura SDTCountriesItem para cargarles su valor.

Nuevo requerimiento: Ranking de países



Se indican los atributos o cálculos con los cuales se cargarán los elementos de la colección :



Como vamos a cargar esta colección basada en el contenido de la tabla COUNTRY, debemos indicar al Data Provider que tiene que recorrer dicha tabla. Para esto usamos la cláusula From, y al lado de ella incluimos el nombre de la transacción cuya tabla base queremos recorrer.

En nuestro caso: From Country

Si la transacción tuviera más de un nivel, para especificar un nivel determinado, asociado a cierta tabla base que queremos navegar, tendríamos que escribir: **nombre de la transacción, punto, nombre del nivel.**

Luego indicamos que al elemento Id lo cargamos con el valor de CountryId, al miembro Name con el valor del atributo CountryName, y al miembro CountryAttractionsQuantity lo queremos cargar con la cantidad de atracciones turísticas que tiene cada país, así que asignamos a este miembro: el resultado de la fórmula inline Count(AttractionName).

Repasemos un concepto ya estudiado: y es que esta fórmula inline definida, navegará la tabla ATTRACTION, por el atributo indicado dentro del paréntesis. Y además, como hay 1 atributo en común en las tablas navegadas por el Data Provider (COUNTRY) y por la fórmula (ATTRACTION), que es CountryId, la fórmula contará las atracciones **del país** navegado por el Data Provider cada vez.

Nuevo requerimiento: Ranking de países

La tabla base del Data Provider es: COUNTRY

De modo que lo que hemos hecho simplemente ha sido: **declarar una tabla a ser navegada por el Data Provider,** y **para cada registro accedido, hemos indicado los valores que deseamos asignar a un ítem nuevo en la colección de países.**

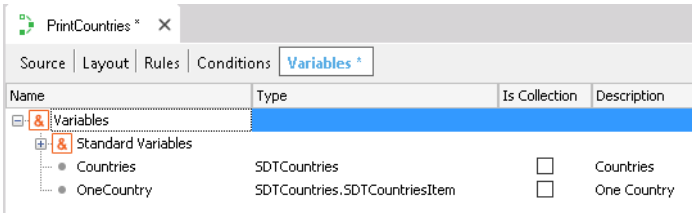
Dado que el Data Provider recorre la tabla COUNTRY, solemos decir **que la tabla base del Data Provider, es COUNTRY:**

El resultado final será que habrán quedado almacenados en la colección en memoria, los datos de todos los países de la base de datos, c/u con su cantidad de atracciones.

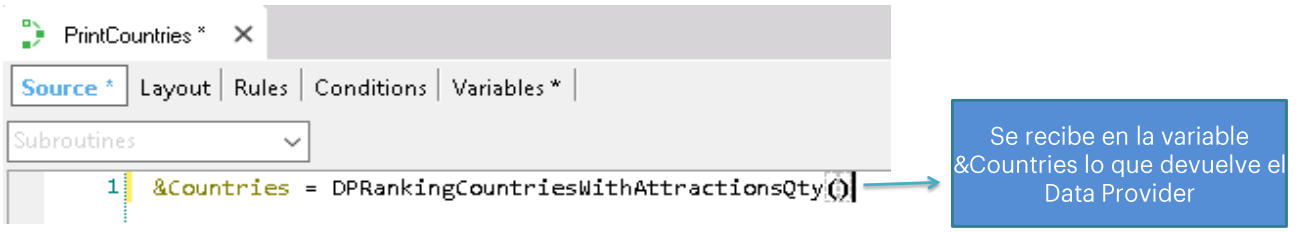
Nuevo requerimiento: Ranking de países

1) Creamos un procedimiento

2) Definimos las variables:



3) En el source del procedimiento, se invoca al Data Provider:



Creamos ahora un objeto Procedimiento para visualizar el contenido de la colección de países. Lo nombramos: "PrintRanking".

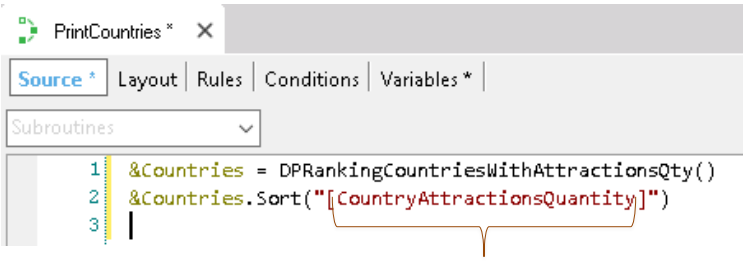
Vamos a la sección de variables del procedimiento y definimos una variable &Countries del tipo SDTCountries.

En el Source del procedimiento, escribimos la asignación que se ve arriba en la diapositiva (a la variable colección Countries, le asignamos el resultado que devuelva el Data Provider que creamos antes).

Con la instrucción que hemos escrito en el source entonces, estamos **invocando** al Data Provider y éste retornará una colección de países, que quedará cargada en la variable &Countries.

Nuevo requerimiento: Ranking de países

Para implementar un ranking, se debe ordenar la colección de mayor a menor por la cantidad de atracciones.... Usamos el método ***Sort:***



Campo del SDT por el cual se desea ordenar la colección

El paréntesis recto dentro de las comillas indica el orden inverso, o sea, de mayor a menor.

Pero recordemos cuál era el requerimiento exacto de la Agencia de Viajes: Era visualizar un ranking de todos los países **ordenados de mayor a menor según la cantidad de atracciones que tienen registradas.**

Por lo tanto, nos está faltando ordenar la colección que obtuvimos cargada. Es decir , ordenar los ítems de la colección de países, antes de ser mostrada, por orden de mayor a menor según la cantidad de atracciones que tienen registradas.

Para resolver esto contamos con el método Sort. La sintaxis es la siguiente:

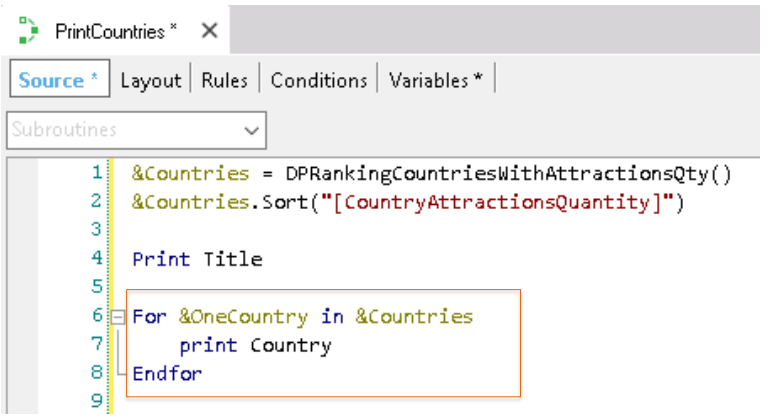
&Countries.Sort(“CountryAttractionsQuantity”)

Pero de esta forma la colección de países quedará ordenada de menor a mayor por la cantidad de atracciones y nosostors necesitamos que se ordene de mayor a menor, ya que queremos implementar un ranking.

Así que para indicar el orden inverso, dentro de las comillas agregamos paréntesis rectos.

Nuevo requerimiento: Ranking de países

Para recorrer la colección de países e imprimir cada elemento en el printblock, utilizamos la estructura *For... in*



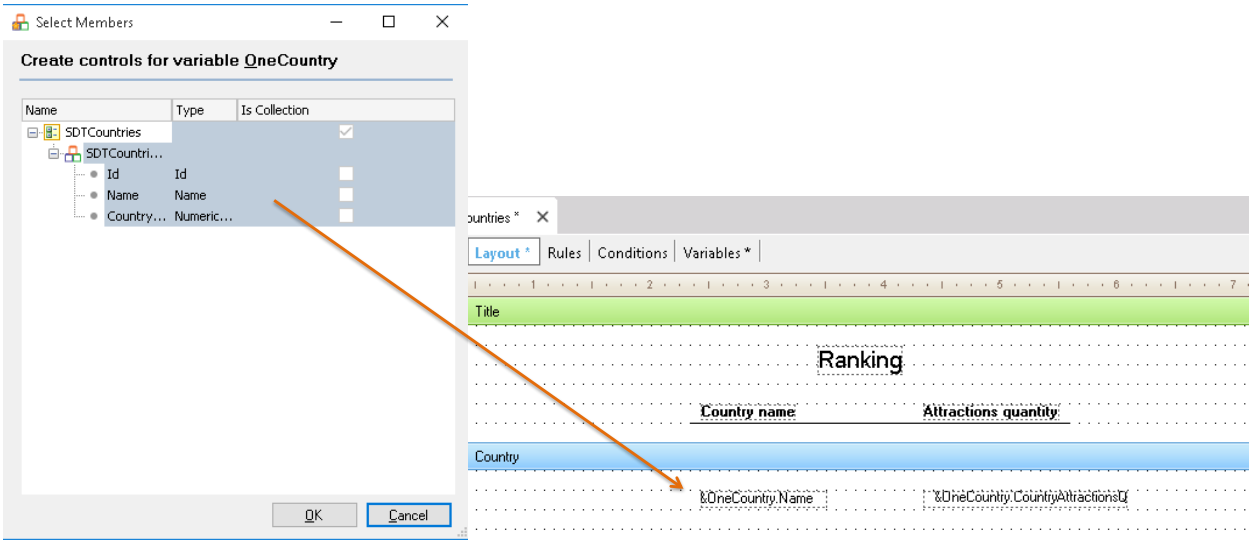
Ahora lo único que nos está faltando es recorrer la colección devuelta por el Data Provider e imprimir para cada ítem de la misma, los datos almacenados.

Para recorrer una colección almacenada en memoria, contamos con el comando **For elemento in Colección.**

Definimos una variable &oneCountry para cargar en ella cada elemento que vayamos iterando de la colección.

Nuevo requerimiento: Ranking de países

Insertamos las variables en el printblock Country....



En la sección “layout” del procedimiento, nombramos a este printblock “Country”, seleccionamos: Insert / Variable y elegimos &oneCountry.

Nuevo requerimiento: Ranking de países

Ranking

Country name	Attractions quantity
France	3
United States	2
Egypt	1
Brazil	1
China	1
Uruguay	0

Y ahora solamente nos resta definir las propiedades necesarias para que se imprima el listado con formato PDF.

Ya hemos configurado las propiedades necesarias, y la regla para definir el formato PDF, así que nuestro requerimiento solicitado está completo.
Seleccionamos run sobre el procedimiento para ver el ranking en ejecución!

Y vemos el listado PDF con todos los países que estaban almacenados en la base de datos, cada uno con su cantidad de atracciones, y el orden en el que se muestran los países es el que nos solicitaron!

Así hemos visto la potencia de los Data Providers para cargar datos en una estructura de datos en memoria, en particular en este caso del tipo colección. Vimos lo sencillo que fue declarar qué queríamos cargar, resolviendo GeneXus todo lo necesario para llevarlo a cabo.

Cláusula Where

```
SDTCountries from Country
Where CountryName <> "Frande"
{
     SDTCountriesItem
     {
          Id = CountryId
          Name = CountryName
          AttractionsQuantity = count(AttractionName)
     }
}
```

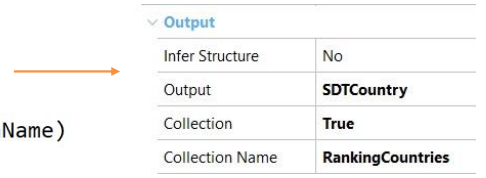
Los Data Providers aceptan opcionalmente la cláusula Where para filtrar, al igual que el comando For each... por ejemplo, si no queremos que el listado incluya a Francia, ¿cómo haríamos?

Podemos agregar la cláusula Where CountryName ... distinto ... de Francia.

Otra forma de implementar este requerimiento...

Name	Type	Description	Is Collection
SDTCountry		SDTCountry	☐
Id	Numeric(4.0)	Id	☐
Name	Character(20)	Name	☐
AttractionsQuantity	Numeric(4.0)	Attractions Quantity	☐

```
SDTCountry from Country
{
     Id = CountryId
     Name = CountryName
     AttractionsQuantity = Count(AttractionName)
}
```



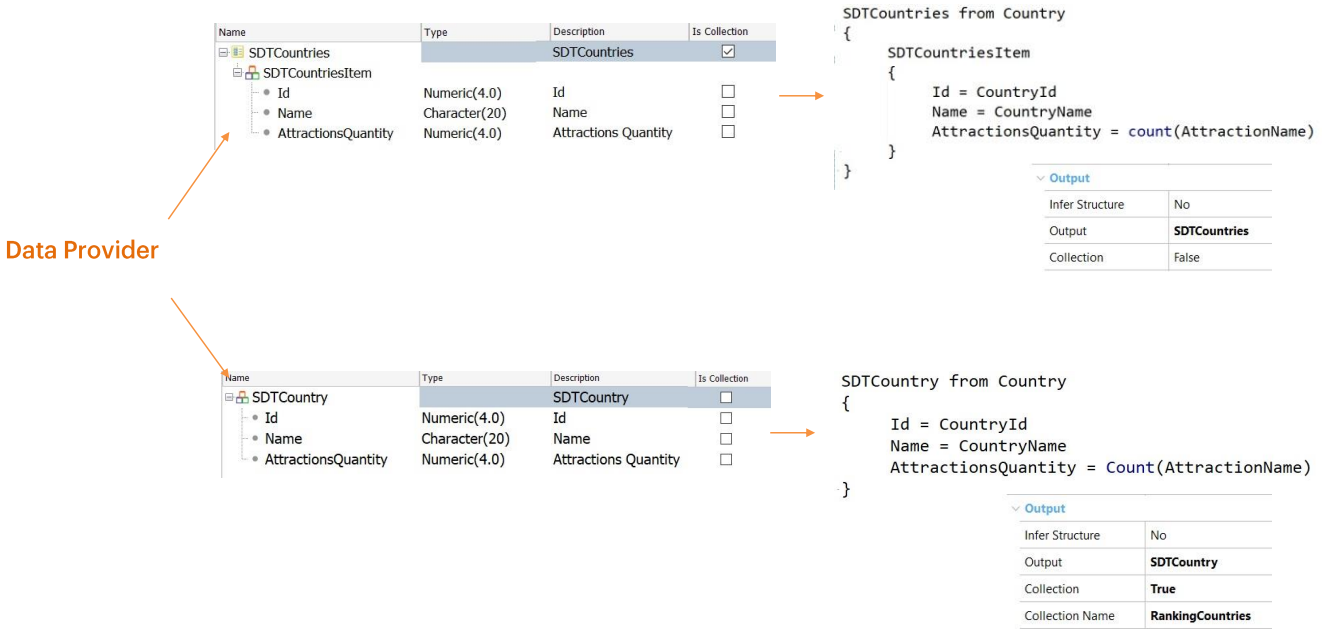
Otra forma de implementar este mismo requerimiento, es a partir del SDT declarado simple y no como colección.
En este caso, la colección deberá ser construida por el Data Provider, y para eso se deberá indicar su propiedad Collection con el valor True.

Al hacer esto le estamos indicando al Data Provider que queremos que nos devuelva una colección de elementos del tipo del SDTCountry.
También observemos que apareció la propiedad Collection Name y que se asignó automáticamente un nombre para la colección.

Al invocar al Data Provider desde el source del procedimiento, debemos definir la variable &CountriesCollection como una colección del tipo de dato estructurado SDTCountry.

Observemos que la sintaxis con que se invoca al Data Provider no cambia, sólo que la variable que recibe el resultado está basada en un SDT simple. Por lo tanto, para que pueda recibir la colección devuelta por el Data Provider debemos declararla como colección marcando el check box IsCollection.

Resumiendo...:



Resumiendo, vemos que tenemos dos formas para que un Data Provider nos devuelva una colección de elementos:

Una de ellas, es definiendo un tipo de datos estructurado del tipo colección y al arrastrarlo al source del Data Provider automáticamente se configura el mismo para retornar una colección de ese tipo,

La otra opción es definiendo un tipo de datos estructurado que no sea una colección y luego mediante las propiedades del Data Provider podemos configurar que el mismo objeto nos arme la colección.

De esta forma hemos visto la potencia de los Data Providers para cargar información en una estructura de datos en memoria. Vimos lo sencillo que fue declarar los datos que queríamos cargar, resolviendo GeneXus todo lo necesario para llevarlo a cabo..

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications