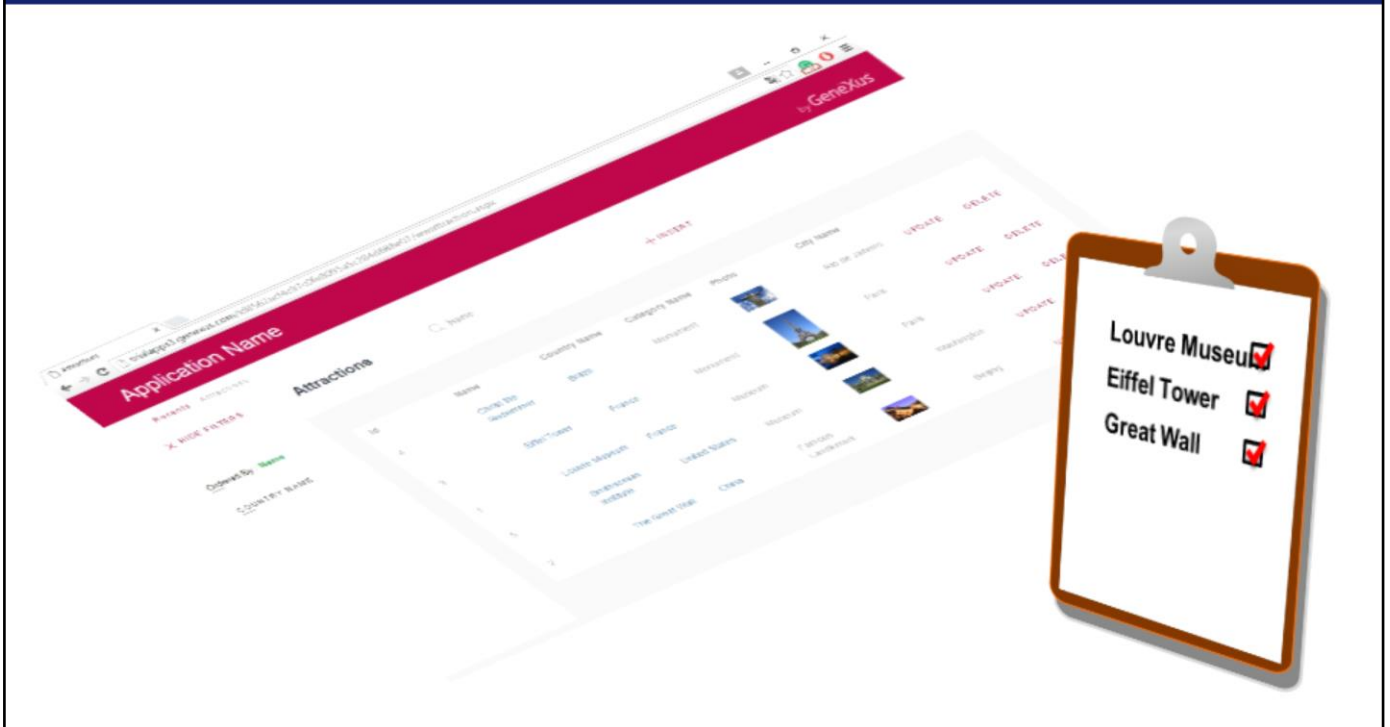


CARGANDO TIPOS DE DATOS COMPUESTOS

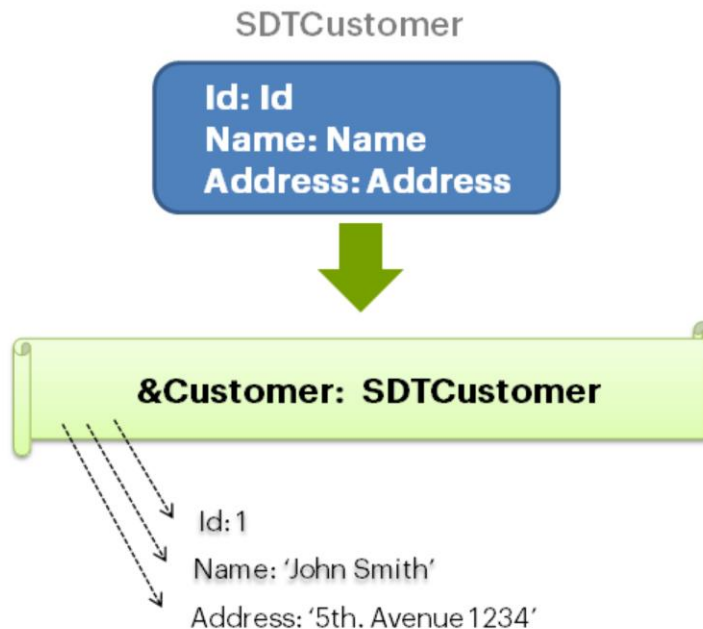
Objeto GeneXus: Data Provider

GeneXus™ 16

En más de una oportunidad, necesitamos almacenar en memoria **una lista de elementos** que tienen el mismo tipo de información pero distintos valores guardados.



Por ejemplo, la agencia de viajes puede necesitar realizar operaciones con un grupo de clientes que cumplan cierto requisito, o nos pueden solicitar procesar información de algunos datos específicos de un conjunto de atracciones turísticas, y eso nos puede implicar tener que cargar estas listas temporalmente en memoria.



Para resolver este tipo de requisito, es necesario crear una estructura en memoria **capaz de almacenar una colección de elementos**.

Ya hemos visto que los tipos de datos estructurados nos permiten definir estructuras que almacenan varios datos correspondientes a **un elemento**.

En esta representación gráfica entonces, estamos almacenando en memoria el identificador, el nombre y la dirección de **un** cliente.

SDTCustomer

Id: 1
Name: John Smith
Address: 5th.
Avenue 1234

← One customer

SDTCustomers

Id: 1
Name: John Smith
Address: 5th. Ave.

Id: 2
Name: Susan
Brown
Address: 7th.Ave.

Id: 3
Name: Robert Hill
Address: 81th. St..

Id: 4
Name: Peter Jensen
Address: St,Paul Rd.

....

↑
A collection of customers

Para almacenar varios elementos con datos de clientes, vimos que es necesario definir un tipo de datos estructurado y marcar que se trata de una colección.

Nuevo requerimiento: Ranking de países

Ranking de países según la cantidad de atracciones turísticas que poseen:



Supongamos ahora que tenemos que implementar un nuevo requerimiento solicitado por la agencia de viajes.

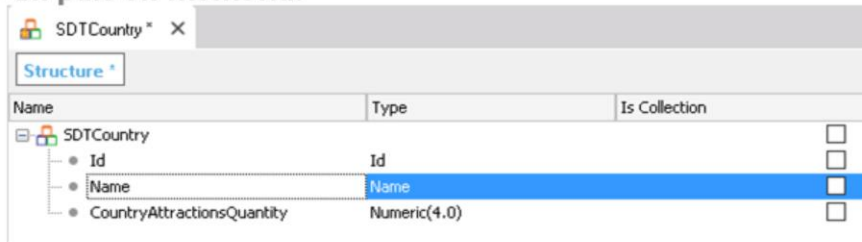
La agencia de viajes desea ofrecer a sus clientes un ranking de países según la cantidad de atracciones turísticas que poseen.

Es decir que debemos mostrar todos los países, ordenados de mayor a menor, en lo que se refiere a la cantidad de atracciones que poseen.

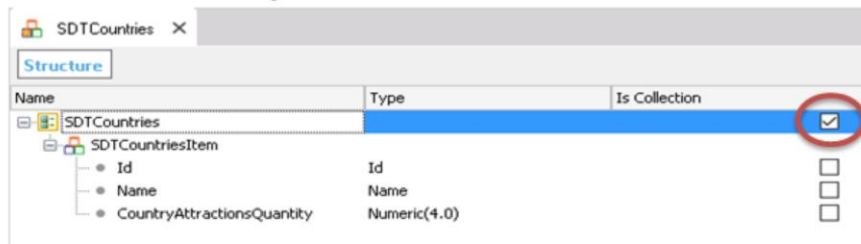
Para poder resolver este requerimiento, necesitamos primero cargar en alguna estructura a todos los países almacenados en la base de datos, cada uno con su correspondiente cantidad de atracciones turísticas. Y luego, tendremos que ordenarlos de mayor a menor por dicha cantidad, para finalmente mostrarlos, ya sea en un web panel o en un listado.

Nuevo requerimiento: Ranking de países

Para cargar un país en memoria:



Para cargar una colección de países en memoria:



Comencemos por crear un nuevo objeto de tipo de datos estructurado y lo nombramos: SDTCountries.

De cada país necesitamos almacenar en memoria su identificador, nombre y la cantidad de atracciones turísticas registradas

Con la primera definición de la diapositiva, estamos representando que se guardará en memoria la información asociada a un solo país.

Pero nosotros necesitamos diseñar un ranking de países, por lo tanto necesitamos almacenar en memoria muchos países. Así que marcamos la casilla "Is collection" para tener una estructura que permita almacenar la información de muchos países (tal como se muestra en la segunda definición de la diapositiva).

La estructura que es colección tiene los mismos miembros que definimos al comienzo, pero agrupados en una subestructura denominada SDTCountriesItem. Esta subestructura se creó automáticamente cuando marcamos que se trata de una colección

Cada ítem almacenará los datos de un país, y la colección al conjunto de países.

Collection structure

Id: 1
Name: Brazil
CountryAttractionsQuantity: 1

Id: 2
Name: France
CountryAttractionsQuantity: 2

Id: 3
Name: China
CountryAttractionsQuantity: 1

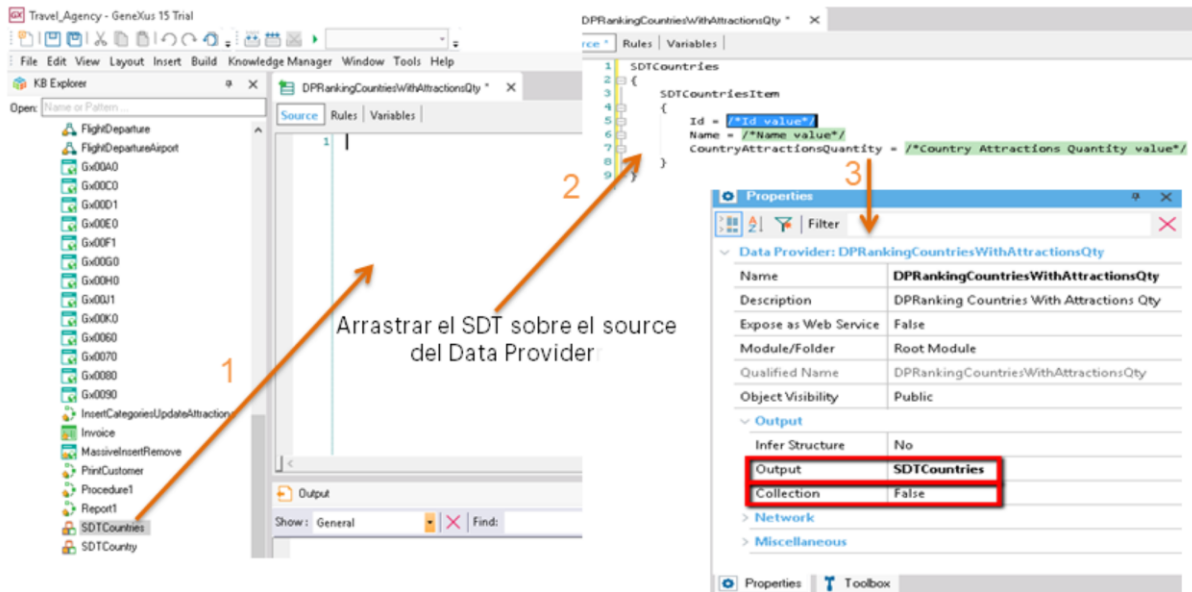
Id: 4
Name: United States
CountryAttractionsQuantity: 1



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 datos 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.**

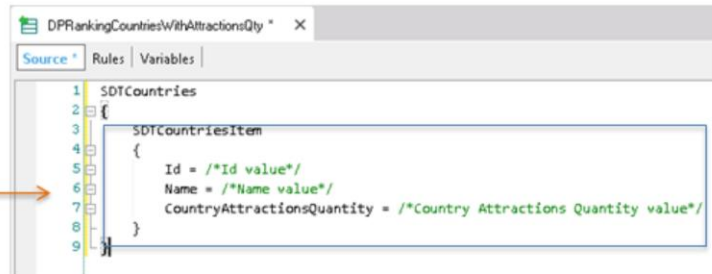
Como el SDTCountries ya es una colección, no es necesario configurar la propiedad Collection con valor True. Esto lo haríamos, si quisiéramos que el DataProvider nos devuelva una colección a partir de un tipo de datos estructurado simple.

Nuevo requerimiento: Ranking de países



Nombre del tipo de datos estructurado.

Subestructura del ítem de la colección.



Estudiamos 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

The screenshot displays the GeneXus IDE interface. The top window, titled 'SDTCountries', shows the 'Structure' tab. It contains a table with the following data:

Name	Type	Is Collection
SDTCountries		<input checked="" type="checkbox"/>
SDTCountriesItem		<input type="checkbox"/>
• Id	Id	<input type="checkbox"/>
• Name	Name	<input type="checkbox"/>
• CountryAttractionsQuantity	Numeric(4.0)	<input type="checkbox"/>

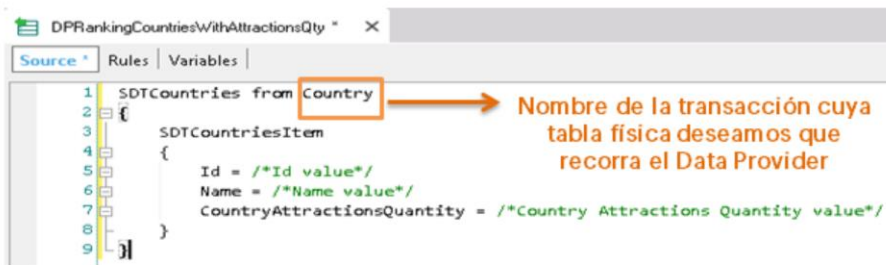
The bottom window, titled 'DPRankingCountriesWithAttractionsQty', shows the 'Source' tab. It contains the following code:

```
1 SDTCountries
2 {
3     SDTCountriesItem
4     {
5         Id = /*Id value*/
6         Name = /*Name value*/
7         CountryAttractionsQuantity = /*Country Attractions Quantity value*/
8     }
9 }
```

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 1 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.

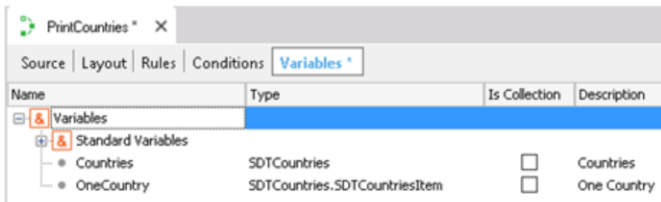
Dado que el Data Provider recorre la tabla COUNTRY, solemos decir **que la tabla base del Data Provider, es COUNTRY**:

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.

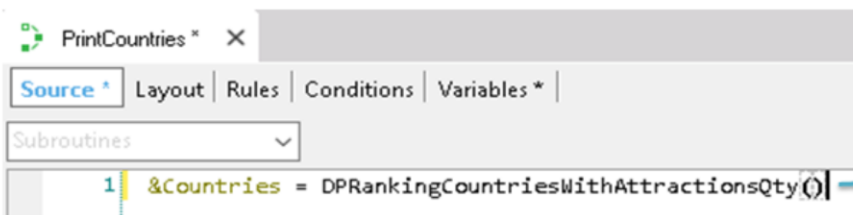
Nuevo requerimiento: Ranking de países

1) Creamos un procedimiento

2) Definimos las variables:



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



Se recibe en la variable &Countries lo que devuelve el 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 nosotros 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*



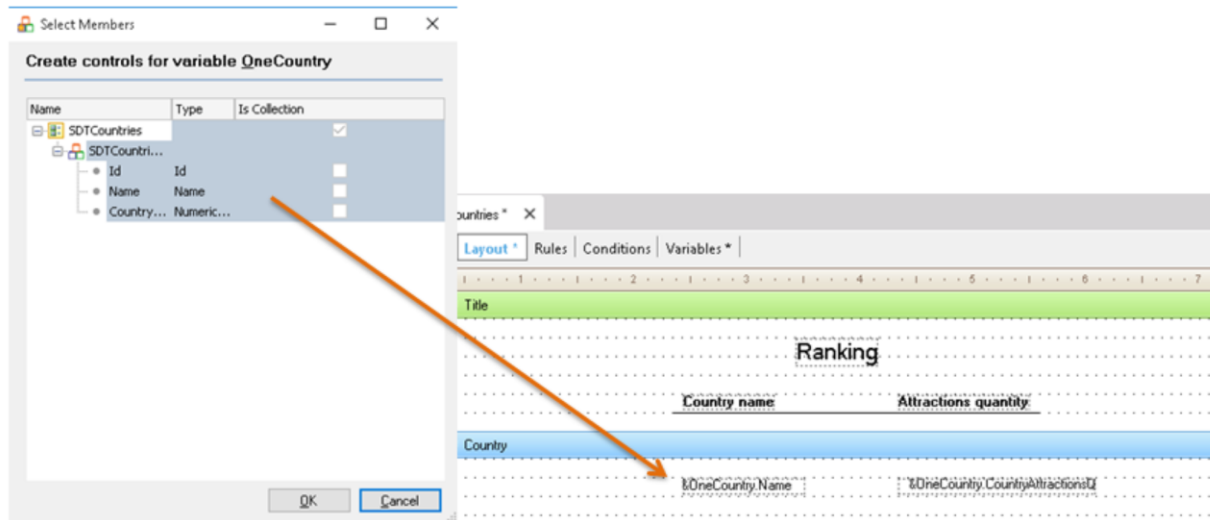
Ahora lo único que nos está faltando es recorrer la colección devuelta por el Data Provider e imprimir para cada ítem de la misma, los datos almacenados.

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.

Vamos a las propiedades del objeto y en “Main program” seleccionamos True.

Luego en “Main object properties” seleccionamos “Call protocol” y elegimos “HTTP”.

Por último tenemos que insertar la regla OutputFile en la sección de reglas, seleccionamos Insert/Rule y elegimos la regla OutputFile.

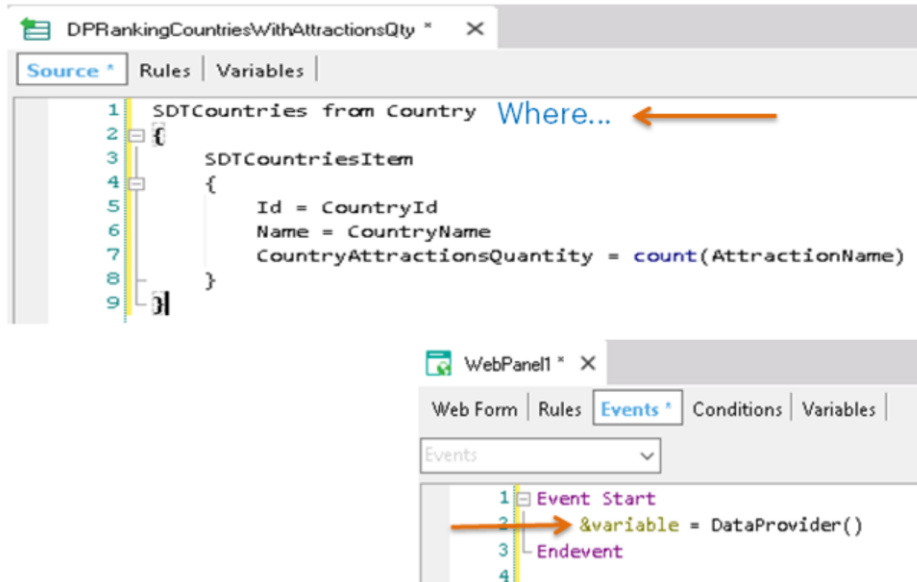
Completamos poniendo el nombre del archivo del listado “Ranking.PDF” y el formato que vamos a usar: “PDF”.

Ahora sí el desarrollo de lo 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.

Nuevo requerimiento: Ranking de países



Los Data Providers aceptan opcionalmente la cláusula where para filtrar, como el comando For each... y más adelante veremos otros ejemplos de uso de Data Providers. Los mismos pueden también ser invocados, en otras secciones de objetos, como por ejemplo en eventos de web panels.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications