

## Tipos de datos compuestos

GeneXus™

Vamos a considerar ahora que la Agencia de Viajes nos solicita un ranking de todos los países, de acuerdo a la cantidad de atracciones turísticas que ofrecen.

## Nuevo requerimiento...

Listado de todos los países, ordenados de mayor a menor por la cantidad de atracciones que tienen registrada..

### Countries Ranking

2	France	3	↓
3	China	2	
4	Egypt	1	
1	Brazil	0	

Necesitamos obtener un listado similar a este, donde podemos ver todos los países, ordenados de mayor a menor por la cantidad de atracciones que tienen registrada.

Como vemos, cada línea corresponderá a un país, del que se mostrará su identificador, su nombre y la cantidad de atracciones turísticas que posee. Pero el problema, aquí, es que tenemos que ordenar esta información de acuerdo a este último valor, que no está en la base de datos. Hay que calcularlo.

Una opción:

Name	Type	Description	Formula
Country	Country	Country	
CountryId	Id	Country Id	
CountryName	Name	Country Name	
CountryAttractionsQty	Numeric(4.0)	Country Attractions Qty	count(AttractionId)
City	City	City	
CityId	Id	City Id	
CityName	Name	City Name	

```

For each Country order (CountryAttractionsQty)
  Print Countries
Endfor

```

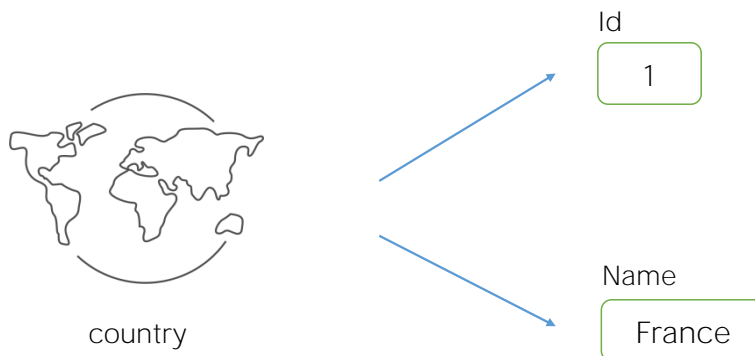
CountryName CountryAttractionsQty

¿Cómo podemos entonces resolver este listado?

Una opción, aplicando los conceptos que ya sabemos, sería definir un atributo fórmula a nivel de la estructura de la transacción Country, y realizar luego un For each, con Country como transacción base, y ordenando de mayor a menor por ese atributo calculado. Recordemos que los paréntesis en el /los atributos indicados en la cláusula Order define un orden inverso, que en este caso sería de mayor a menor.

Esta solución es totalmente válida, pero es válido también resolver un requerimiento sin la necesidad de agregar nuevos atributos a una transacción solamente con el fin de resolver una determinada consulta.

## Tipos de datos simples



Así que vamos a resolver este requerimiento de otra manera, agregando nuevos conceptos que nos serán de mucha utilidad para casos más complejos.

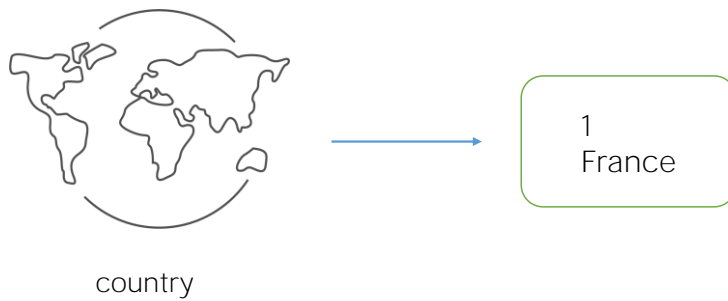
Comencemos a hablar entonces de los Tipos de datos estructurados.

Hasta ahora hemos empleado siempre tipos de datos simples. Hemos definido atributos y dominios de tipo Numeric, de tipo Character, Date, Image, etc.

Si, por ejemplo, queremos ahora guardar en variables el identificador y nombre de un país, entonces necesitamos dos variables.

Pero vamos a ver a continuación, que contamos también con la posibilidad de tener tipos de datos compuestos.

## Tipos de datos compuestos o estructurados




Un tipo de datos compuesto solamente podrá ser asignado a una variable, **nunca a un atributo**.

Un tipo de datos compuesto permite almacenar varios datos juntos en una sola variable. Para decirlo de una manera simple, es como juntar varias variables simples y agruparlas bajo un nombre.

Un tipo de datos estructurado se crea a través de un objeto GeneXus de este tipo. El tipo de dato creado solamente podrá luego ser asignado a una variable, nunca a un atributo.

## Objeto GeneXus: Structured Data Type (SDT)

Structure *			
Name	Type	Description	Is Collection
 SDTCountry <ul style="list-style-type: none"> <li>• Id</li> <li>• Name</li> </ul>		SDTCountry	<input type="checkbox"/>
	Numeric(4.0)	Id	<input type="checkbox"/>
	Character(20)	Name	<input type="checkbox"/>

Vamos entonces a crear un objeto de tipo SDT y le ponemos como nombre SDTCountry. Aclaremos que le ponemos el término SDT delante porque no pueden existir dentro de la misma KB una transacción y un tipo de dato estructurado con el mismo nombre.

Recordemos que ya tenemos la transacción Country.

Por ahora, queremos solamente guardar el Identificador y el nombre de los países así que definimos esos ítems, o esos miembros en la estructura:

Id, de tipo Numérico

Y Name, de tipo Character

Para esta definición que acabamos de hacer, GeneXus creó el tipo de dato SDTCountry, por lo tanto ya podemos comenzar a crear variables basadas en este tipo de dato.

La pregunta que nos hacemos ahora es ¿cómo podemos cargar valores en una variable estructurada?

## Carga de un SDT

### 1) Carga manual

```
&CountryItem.Id = 1  
&CountryItem.Name = "France"
```

### 2) Carga dentro de un For each

```
For each Country  
  Where CountryName = "France"  
  &CountryItem.Id = CountryId  
  &CountryItem.Name = CountryName  
  print printblock1  
Endfor
```

Como primer ejemplo vamos a cargar los datos de Francia, y lo haremos en forma manual.

Al digitar la variable `&CountryItem`, y presionar la tecla del punto, ya podemos ver los ítems que componen el tipo de dato.

Vamos a cargar entonces el Id 1 que corresponde a Francia, y el nombre "Francia". De esa forma cargamos la variable en forma manual.

Otra opción sería a través de un For each posicionándonos en el registro correspondiente a Francia, y cargamos entonces la variable `&CountryItem` con el valor de `CountryId` para el ítem `Id`, y con el valor de `CountryName` para el ítem `Name`.

## Definición de colecciones

Name	Type	Description	Is Collection
SDTCountry		SDTCountry	<input type="checkbox"/>
CountryId	Attribute:CountryId	Country Id	<input type="checkbox"/>
CountryName	Attribute:CountryName	Country Name	<input type="checkbox"/>
City		City	<input checked="" type="checkbox"/>
CityItem			
CityId	Attribute:CityId	City Id	<input type="checkbox"/>
CityName	Attribute:CityName	City Name	<input type="checkbox"/>

SDT colección

Variable colección

Variables *			
Name	Type	Is Collection	Description
Variables			
Standard Variables			
CountryItem	SDTCountry	<input type="checkbox"/>	Country Item

Veamos ahora que cuando la estructura que queremos crear coincide totalmente, o en parte, con la estructura de una transacción, podemos arrastrar dicha transacción sobre la estructura del SDT, sin temor a que se genere ningún tipo de ambigüedad, ya que GeneXus puede distinguir que una cosa son los atributos de la transacción, y otra son los ítems de un SD, aunque sus nombres coincidan.

Recordemos que un tipo de dato estructurado puede ser asignado solamente a variables y no a atributos.

Observemos también que la estructura de un SDT puede llegar a ser muy compleja. Aquí por ejemplo cada país tiene una colección de ciudades, y se ve claramente que los tipos de datos de los miembros del SDT provienen de los atributos.

¿Y si necesitamos guardar una colección de países? ¿Cómo podemos hacerlo? Tenemos dos opciones:

Una es definiendo nuestro SDT como una colección, y para eso alcanza con marcar esta casilla que dice IsCollection.

De esta forma, al grabar los cambios, GeneXus creará el tipo de dato SDTCountry, para la colección, y el tipo de dato SDTCountry.SDTCountryItem para el elemento de la colección.

Otra forma es dejar nuestro SDT como lo hemos definido originalmente y marcar la colección a nivel de la variable definida.

En el próximo video hablaremos detenidamente sobre las variables colección.



*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)