

# Tipos de dados compostos

*GeneXus™*

Vamos considerar agora que a Agência de Viagens nos solicita um ranking de todos os países, de acordo com a quantidade de atrações turísticas que oferecem.

## Novo requisito ...

Listar todos os países, ordenados do maior para o menor pela quantidade de atrações que têm registradas.

### Countries Ranking

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

Precisamos obter uma lista similar a esta, onde podemos ver todos os países, ordenados do maior para o menor pela quantidade de atrações que têm registradas.

Como vemos, cada linha corresponderá a um país, do qual se mostrará seu identificador, seu nome e a quantidade de atrações turísticas que possui. Mas o problema aqui, é que temos que ordenar esta informação de acordo com este último valor, que não está na base de dados. Temos que calculá-lo.

## Uma opção :

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

```

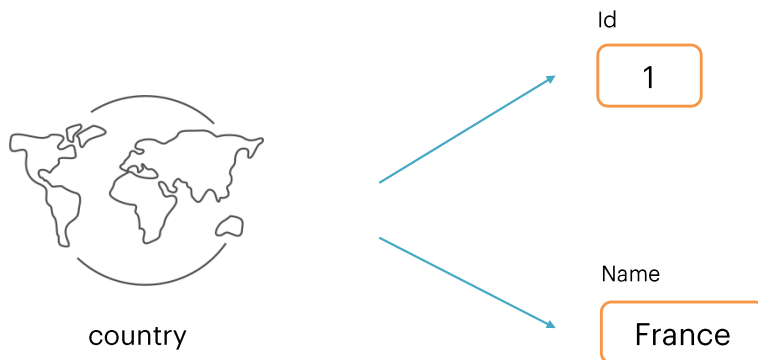
CountryName CountryAttractionsQty

Como podemos então resolver esta lista?

Uma opção, aplicando os conceitos que já sabemos, seria definir um atributo fórmula no nível da estrutura da transação Country, e depois realizar um For each, com Country como transação base, e ordenando, do maior para o menor, por esse atributo calculado. Recordemos que os parênteses no/nos atributos indicados na cláusula Order, define uma ordem inversa, que neste caso seria do maior para o menor

Esta solução é totalmente válida, mas é válido também resolver um requisito sem a necessidade de adicionar novos atributos a uma transação somente com a finalidade resolver uma determinada consulta.

## Tipos de dados simples



Então vamos resolver esta solicitação de outra forma, adicionando novos conceitos que serão muito úteis para casos mais complexos. Começemos então, a falar dos Tipos de dados estruturados.

Até agora temos empregado sempre tipos de dados **simples**. Definimos atributos e domínios de tipo Numeric, de tipo Character, Date, Image, etc.

Se, por exemplo, queremos agora guardar em variáveis o identificador e nome de um país, então precisamos de duas variáveis.

Mas vamos ver a seguir, que também temos a possibilidade de ter tipos de dados **compostos**.

## Tipos de dados compostos o estruturados



Um tipo de dados composto somente poderá ser atribuído a uma variável, nunca a um atributo.

Um tipo de dados composto permite armazenar vários dados juntos em uma única variável. Para colocar de uma forma simples, é como juntar várias variáveis simples e agrupá-las sob um nome.

Um tipo de dados estruturado é criado através de um objeto GeneXus deste tipo. O tipo de dado criado somente poderá então ser atribuído a uma variável, nunca a um atributo.

## Objeto GeneXus: Structured Data Type (SDT)

Structure *			
Name	Type	Description	Is Collection
SDTCountry		SDTCountry	<input type="checkbox"/>
• Id	Numeric(4.0)	Id	<input type="checkbox"/>
• Name	Character(20)	Name	<input type="checkbox"/>

Vamos então criar um objeto de tipo SDT e colocamos como nome SDTCountry. Esclarecemos que lhe colocamos o termo SDT na frente porque não podem existir dentro da mesma KB uma transação e um tipo de dado estruturado com o mesmo nome.

Recordemos que já temos a transação Country.

Por enquanto, queremos somente salvar o Identificador e o nome dos países, então definimos esses itens, ou esses membros na estrutura:

- Id, de tipo Numérico
- E Name, de tipo Character

Para esta definição que acabamos de fazer, GeneXus criou o tipo de dado SDTCountry, portanto já podemos começar a criar variáveis baseadas neste tipo de dado.

A pergunta que fazemos agora é: como podemos carregar valores em uma variável estruturada?

## Carregando um SDT

### 1) Carregamento manual

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

### 2) Carregamento dentro de um For each

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

Como primeiro exemplo, vamos carregar os dados da França, e faremos isso de forma manual.

Ao digitar a variável &CountryItem, e pressionar a tecla do ponto, já podemos ver os itens que compõem o tipo de dado.

Vamos carregar então o Id 1 que corresponde à França, e o nome "França". Dessa forma, carregamos a variável de forma manual.

Outra opção seria através de um For each posicionando-nos no registro correspondente à França, e carregamos, então, a variável &CountryItem com o valor de CountryId para o item Id, e com o valor de CountryName para o item Name.

## Definindo coleções

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 coleção

Variáveis coleção

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

Vejamos agora que quando a estrutura que queremos criar coincide totalmente, ou em parte, com a estrutura de uma transação, podemos arrastar essa transação sobre a estrutura do SDT, sem medo de gerar qualquer tipo de ambiguidade, já que GeneXus pode distinguir que uma coisa são os atributos da transação, e outra são os itens de um SDT, embora seus nomes coincidam.

Recordemos que um tipo de dado estruturado pode ser atribuído somente a variáveis e não a atributos.

Observemos também que a estrutura de um SDT pode tornar-se muito complexa! Aqui, por exemplo, cada país tem uma coleção de cidades, e se vê claramente que os tipos de dados dos membros do SDT provêm dos atributos.

E se precisamos guardar uma coleção de países? Como podemos fazer isso? Temos duas opções:

Uma é definindo nosso SDT como uma coleção, e para isso basta marcar esta caixa que diz IsCollection.

Desta forma, ao gravar as alterações, GeneXus criará o tipo de dado SDTCountry, para a coleção, e o tipo de dado SDTCountry.SDTCountryItem para o elemento da coleção.

Outra forma é deixar o nosso SDT como o definimos originalmente e marcar a coleção no nível da variável definida.

No próximo vídeo, falaremos detalhadamente sobre as variáveis coleção.



# GeneXus™

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