

TIPO DE DADOS COMPOSTOS

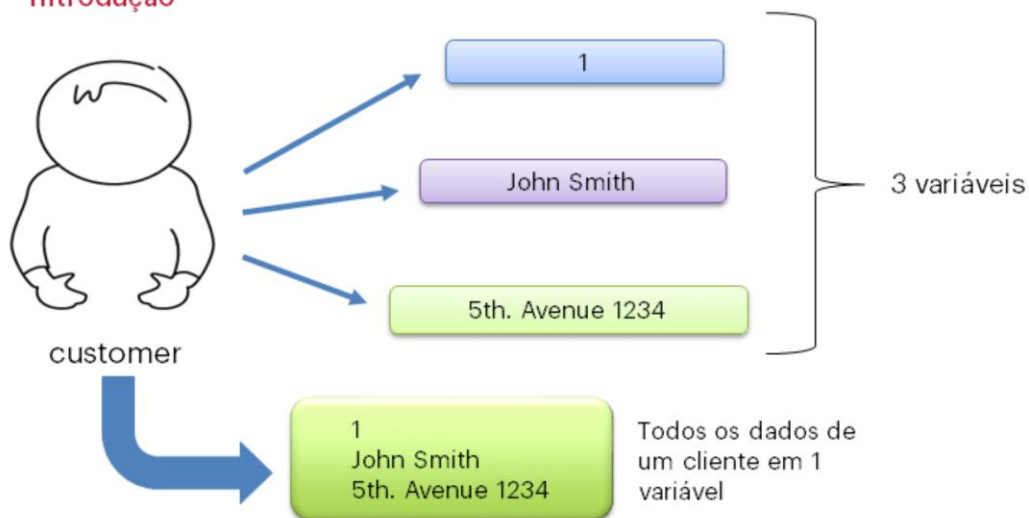
GeneXus object: Structured Data Type

GeneXus™ 16

Até agora, sempre usamos tipos de dados **simples**. Nós definimos atributos e domínios do tipo Numérico, do tipo Caractere, e dos tipos de Dados e Imagem, entre outros.

Agora veremos que há casos em que seria útil ter a possibilidade de ter tipos de dados **compostos**.

Introdução

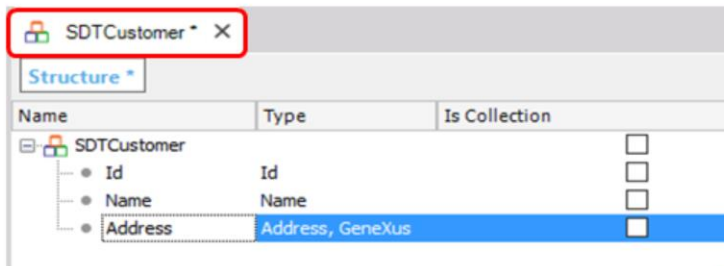


Definiremos um tipo de dados estruturado (SDT) e, em seguida, uma variável deste tipo

Se, por exemplo, precisássemos guardar em memória os dados relativos a um cliente em um objeto específico, teríamos duas opções:

1. Definindo uma variável individual para armazenar o ID, outra variável individual para armazenar o nome, outra para o endereço e assim por diante.
2. Ou, GeneXus também oferece a possibilidade de **armazenar vários pedaços de dados juntos em uma única variável**. Nesta opção, precisamos definir um tipo de dados especial conhecido como **tipo de dados composto** ou também **tipo de dados estruturado (SDT)**, e então criar uma variável com este tipo de dados.

Definindo um SDT



customer



&OneCustomer: SDTCustomer

```
&OneCustomer.Id = 1  
&OneCustomer.Name = 'John Smith'  
&OneCustomer.Address = '5th. Avenue 1234'
```

Sintaxe de atribuição de dados fixos

Apenas as variáveis podem ser do tipo SDT
Os atributos NÃO!

Quando definimos o SDT adicionamos cada membro ou o nome dos dados que queremos armazenar em relação ao cliente, com o seu tipo de dados correspondente.

Podemos atribuir esta definição que produzimos de um tipo de dados composto, como tipo de dados de uma **variável** que vamos definir em qualquer objeto de GeneXus.

Nós não podemos usar um tipo de dados estruturado para definir um atributo porque atributos só podem armazenar dados simples.

O slide mostra a sintaxe para atribuir, para a variável &OneCustomer (criada como tipo de dados estruturado sob o nome SDTCustomer), dados específicos correspondentes a **um cliente**.

Definindo um SDT, outra forma

The top screenshot shows the 'Structure' tab for the SDT 'SDTCustomer2'. It lists the following attributes:

Name	Type	Is Collection
CustomerId	Attribute:CustomerId	<input type="checkbox"/>
CustomerName	Attribute:CustomerName	<input type="checkbox"/>
CustomerLastName	Attribute:CustomerLastName	<input type="checkbox"/>
CustomerAddress	Attribute:CustomerAddress	<input type="checkbox"/>
CustomerPhone	Attribute:CustomerPhone	<input type="checkbox"/>
CustomerEmail	Attribute:CustomerEmail	<input type="checkbox"/>

The bottom screenshot shows the 'Variables' tab for the same SDT. It displays 'Standard Variables' with the following details:

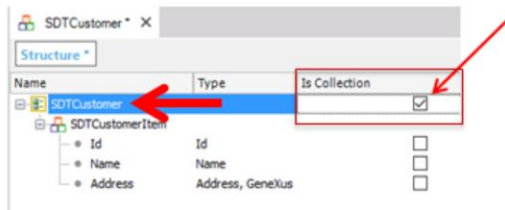
Name	Type	Is Collection	Description
OneCustomer	SDTCustomer	<input type="checkbox"/>	One Customer
AnotherCustomer	SDTCustomer2	<input type="checkbox"/>	Another Customer

Em vez de começar a definir membros do SDT um por um, podemos arrastar a transação Customer a partir do Root Module e soltá-lo sobre o SDT estrutura que estávamos definindo.

Os membros da SDTCustomer2 são criados automaticamente, com os mesmos nomes que os atributos da transação Customer e seus tipos de dados.

Definindo um SDT coleção

- Se quisermos armazenar em memória os dados de vários clientes:



- Podemos definir:
 - uma variável do tipo **SDTCustomer** → coleção E.g: &CustomersList
 - outra variável do tipo **SDTCustomer.SDTCustomerItem** → 1 elemento da coleção E.g: &OneCustomer
- Quando estudarmos Data Providers, veremos como carregar dados do BD em uma variável do tipo SDT (simples ou coleção).
- Depois de carregadas as variáveis do tipo SDT (simples ou coleção), as utilizamos de diferentes maneiras de acordo com a necessidade.

Embora até agora mostramos o uso do SDT para armazenar na memória temporária os dados relativos a **um** cliente, nós podemos modificar facilmente a definição, para armazenar os dados de **muitos** clientes. Marcando a caixa Is Collection localizada à direita do nome SDTCustomer, estaremos definindo que o **SDT** armazena uma **coleção** de elementos da estrutura definida (em vez de um único elemento como antes). Cada item na coleção irá armazenar os dados de um cliente e a coleção irá armazenar o grupo de clientes.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications