

TIPO DE DATOS COMPUESTOS

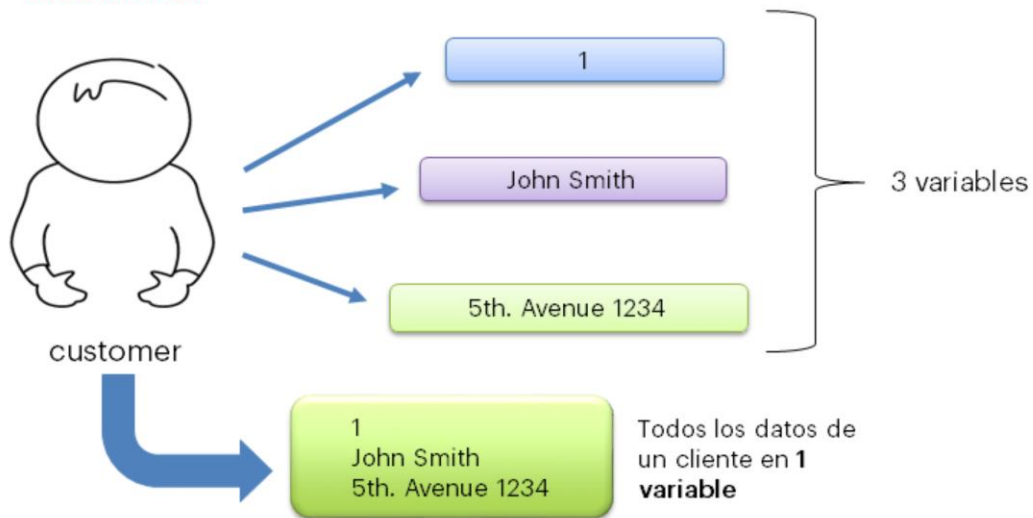
GeneXus object: Structured Data Type

GeneXus™ 16

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

Vamos a ver a continuación, que hay casos en que sería útil contar con la posibilidad de tener tipos de datos **compuestos**.

Introducción

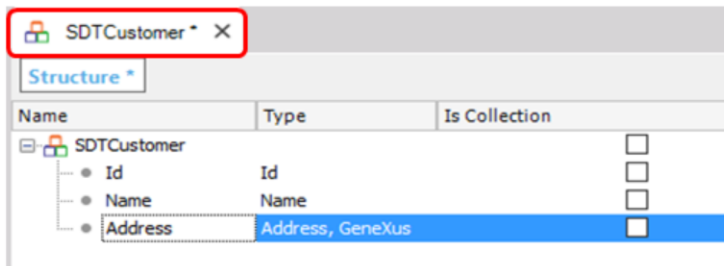


Definiremos un tipo de datos estructurado (SDT) y luego una variable de dicho tipo

Si por ejemplo necesitáramos guardar en memoria los datos de un cliente en determinado objeto, tenemos 2 posibilidades:

1. Definimos una variable individual para guardar el id, otra variable individual para guardar el nombre, otra más para la dirección, etc.
2. O sino, GeneXus nos brinda también la posibilidad de **almacenar varios datos juntos en una sola variable**. Para resolverlo de esta última forma, tenemos que definir un tipo de datos especial, llamado **tipo de datos compuesto o tipo de datos estructurado** (en inglés: Structured Data Type o SDT) y luego crear una variable de ese tipo de datos

Definiendo un SDT



customer



&OneCustomer: SDTCustomer

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

Sintaxis de asignación de datos fijos

Sólo las variables pueden ser del tipo SDT
Los atributos NO!

En la definición del SDT agregamos cada miembro o nombre de dato que nos interesa guardar del cliente, con su tipo de datos correspondiente.

Esta definición que hicimos de un tipo de datos compuesto podemos asignarla como tipo de datos de una variable que definamos en cualquier objeto GeneXus.

No podemos usar un tipo de datos estructurado para definir un atributo, ya que los atributos pueden almacenar solamente datos simples.

La diapositiva muestra la sintaxis para asignar a la variable &OneCustomer (creada del tipo de datos estructurado de nombre SDTCustomer), ciertos datos correspondientes a **un cliente**.

Definiendo un SDT, otra forma

The first screenshot shows the 'Structure' tab of the IDE. It displays a tree view for 'SDTCustomer2' with the following members:

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 second screenshot shows the 'Variables' tab. It displays a tree view for 'Variables' with the following members:

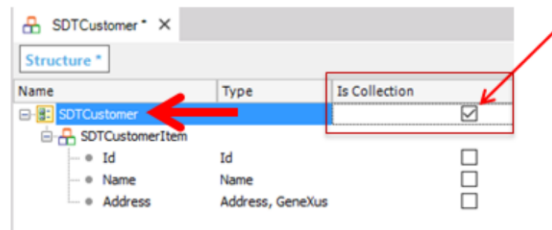
Name	Type	Is Collection	Description
OneCustomer	SDTCustomer	<input type="checkbox"/>	One Customer
AnotherCustomer	SDTCustomer2	<input type="checkbox"/>	Another Customer

En lugar de empezar a definir los miembros del SDT uno a uno, arrastramos desde el Root Module a la transacción Customer sobre la estructura del SDT que estábamos definiendo y soltamos.

Se crean automáticamente los miembros del SDTCustomer2, con los mismos nombres de los atributos de la transacción Customer y sus tipos de datos.

Definiendo un SDT colección

- Si queremos guardar en memoria los datos de **varios** clientes:



- Podremos definir:
 - una variable de tipo **SDTCustomer** → colección Ej: &CustomersList
 - otra variable de tipo **SDTCustomer.SDTCustomerItem** → 1 elemento de la colección Ej: &OneCustomer
- Cuando estudiemos Data Providers veremos cómo cargar datos de la BD en una variable de tipo SDT (simple o colección).
- Luego de cargadas las variables de tipo SDT (simple o colección), las utilizamos de variadas formas según la necesidad.

Si bien hasta ahora hemos mostrado el uso de SDT para guardar en memoria temporal los datos de **un** cliente, vemos que muy fácilmente podemos modificar su definición para guardar los datos de **muchos** clientes. Marcando la casilla Is Collection ubicada a la derecha del nombre SDTCustomer estaremos definiendo que **el SDT** almacenará una **colección** de elementos de la estructura definida (en lugar de un solo elemento como ocurría antes). Cada ítem de la colección almacenará los datos de un cliente y la colección al conjunto de clientes.



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications