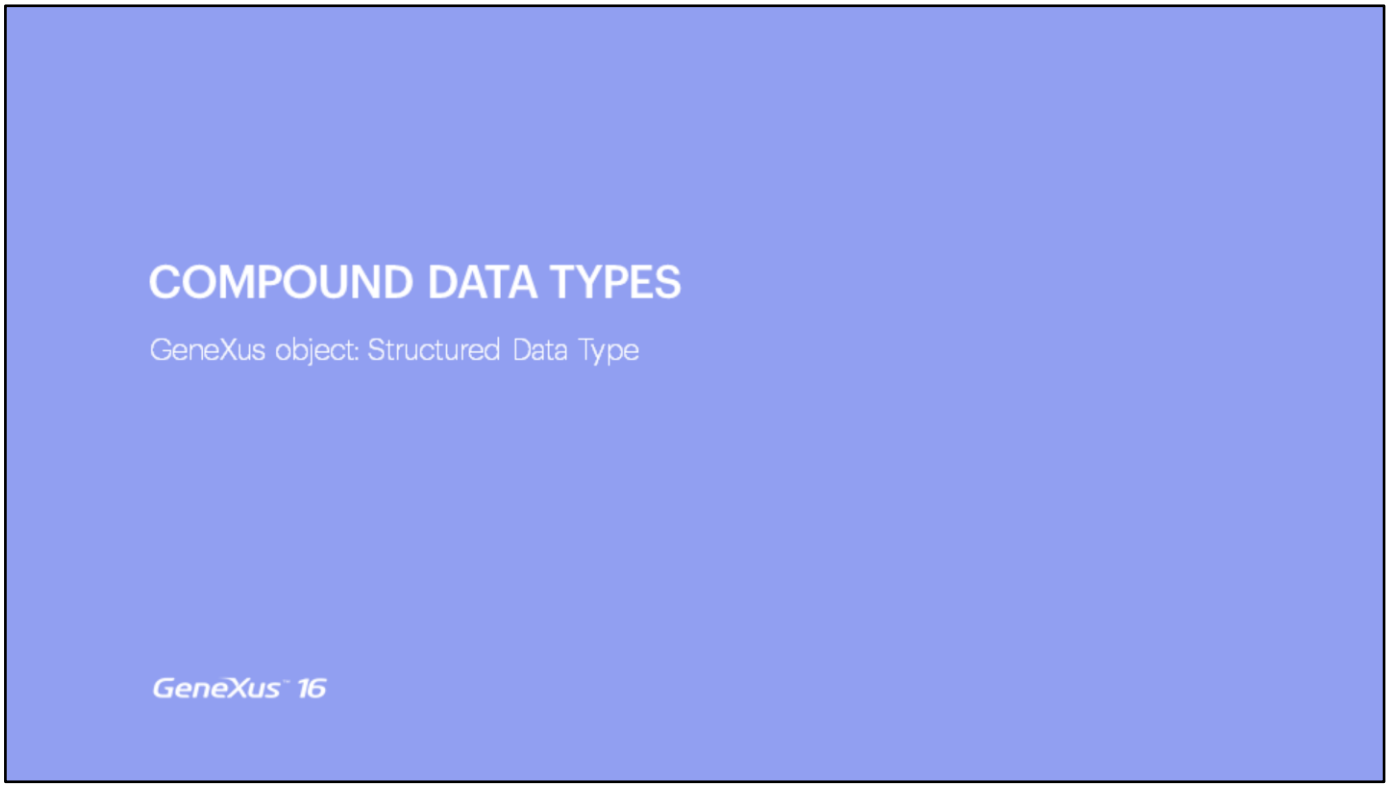
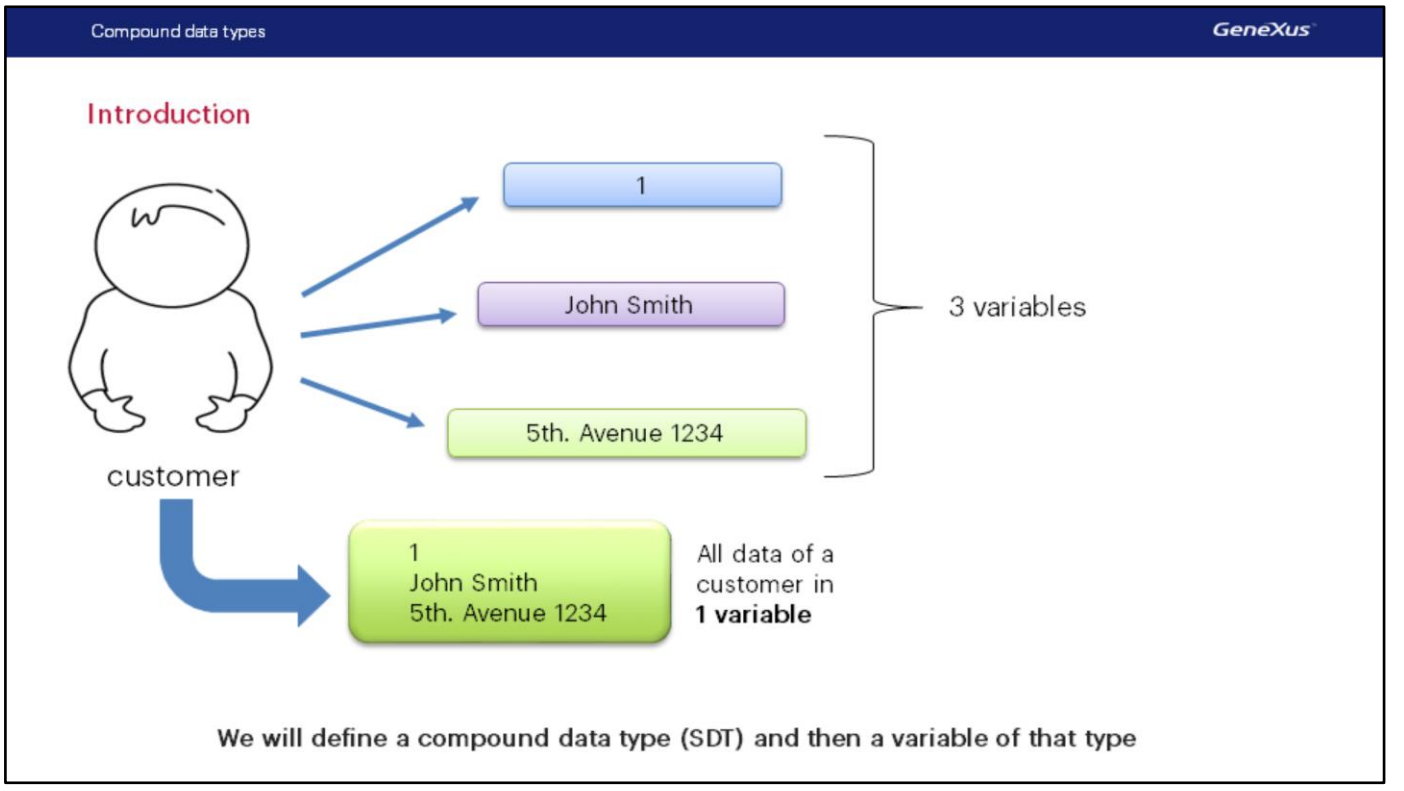


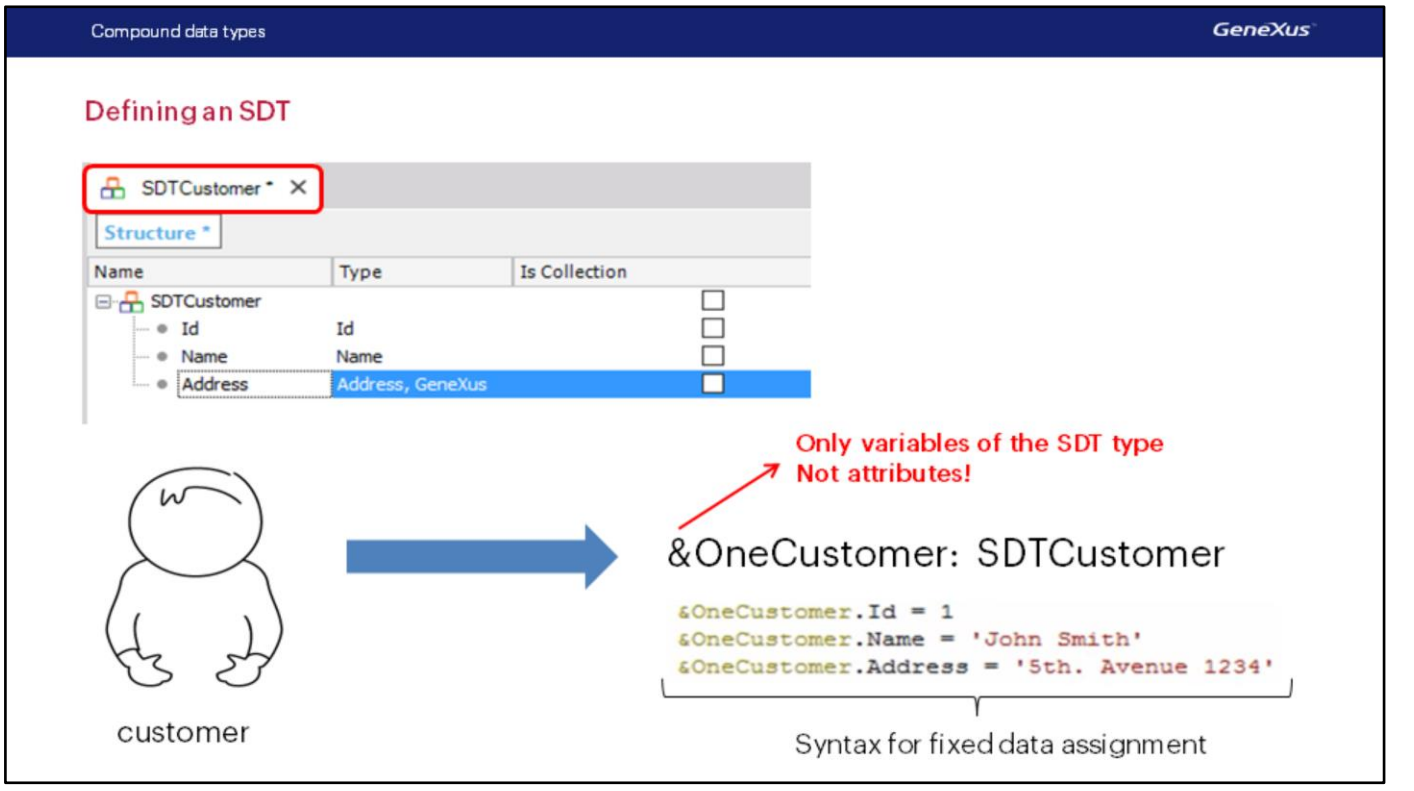
So far, we have always used **simple** data types. We have defined attributes and domains of the Numeric type, of the Character type, and of the Data and Image types, among others.

We will now see that there are cases where it would be useful to have the possibility of having **compound** data types.



If, for instance, we needed to save in memory the data relative to a customer in a specific object, we would have two options:

1. Defining an individual variable to store the ID, another individual variable to store the name, another one for the address, and so on.
2. Or, GeneXus also offers the possibility of **storing several pieces of data together in a single variable**. In this option we need to define a special data type known as **compound data type** or also **structured data type** (SDT), and then create a variable with this data type.

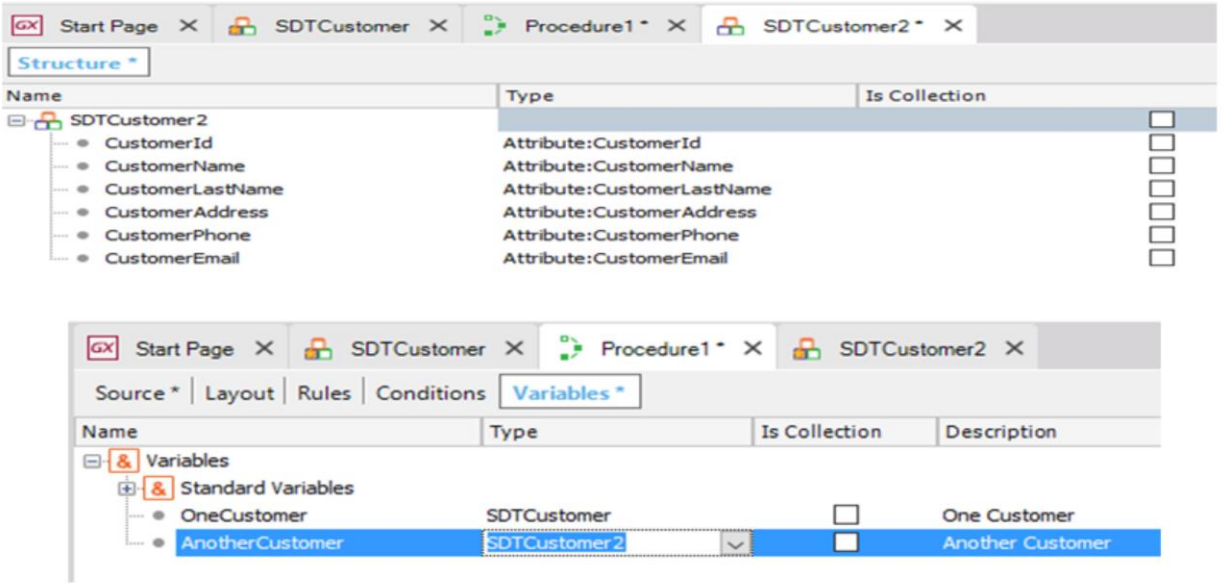


When we define the SDT we add each member or name of the data we want to store in relation to the client, with its corresponding data type.

We can assign this definition we produced of a compound data type, as data type of a **variable** we will define in any GeneXus object.
We cannot use a structured data type to define an attribute because attributes may only store simple data.

The slide shows the syntax for assigning, to the variable &OneCustomer (creates as structured data type under the name SDTCustomer), specific data corresponding to **a client.**

Another way to define an SDT

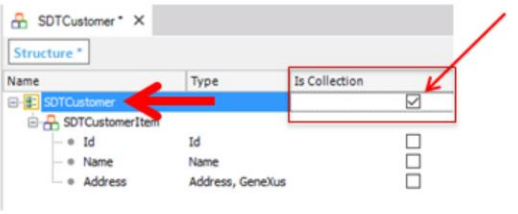


Instead of starting to define the SDT's members one by one, we can drag the Customer transaction from the Root Module and drop it on the SDT structure that we were defining.

The members of the SDTCustomer2 are automatically created, with the same names as the attributes in the Customer transaction and their data types.

Defining a collection SDT

- If we want to save the data of **several** customers in memory:



- We can define:
  - a variable of the SDTCustomer type collection E.g: &CustomersList
  - another variable of the SDTCustomer.SDTCustomerItem type → 1 element of the collection E.g: &OneCustomer
- When we study Data Providers we will see how to load data of the DB in a variable of the SDT type (simple or collection).
- After the SDT type variables (simple or collection) have been loaded, we will use them in several ways, as needed.

Although so far we have shown the use of SDT to store in temporary memory the data relative to **one** client, we can easily modify the definition, to store the data of **many** clients. By checking the Is Collection box located to the right of the SDTCustomer name, we will be defining that the **SDT** stores a **collection** of elements of the structure defined (instead of a single element as before). Each item in the collection will store the data of one client and the collection will store the group of clients.

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications