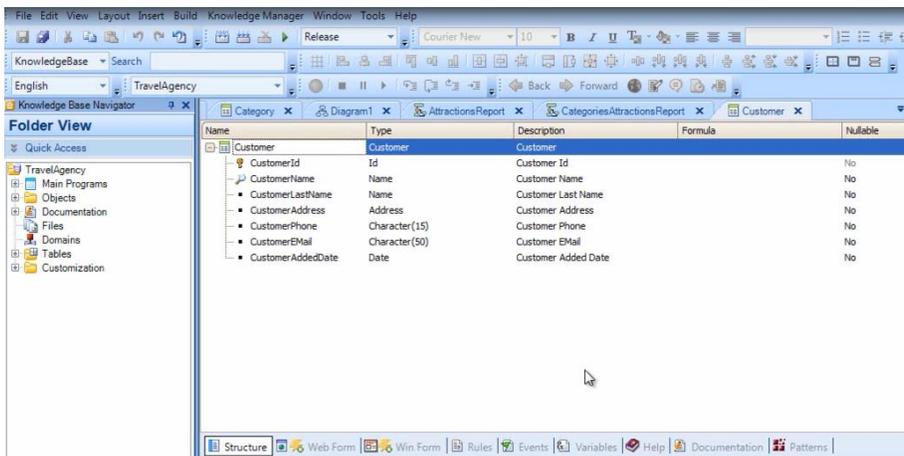


## Tipos de datos compuestos

### Compound data types

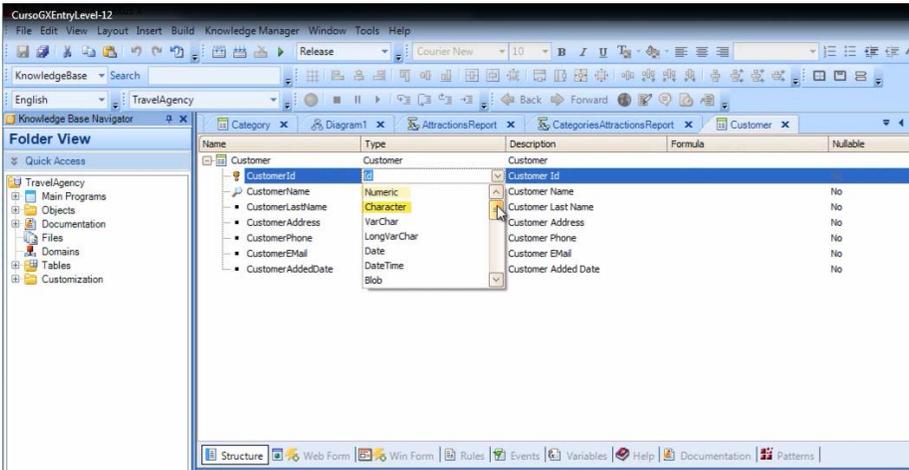


The screenshot shows the Knowledge Manager interface with a table defining the 'Customer' data type. The table has columns for Name, Type, Description, Formula, and Nullable. The 'Customer' type is defined as a 'Customer' with the following attributes:

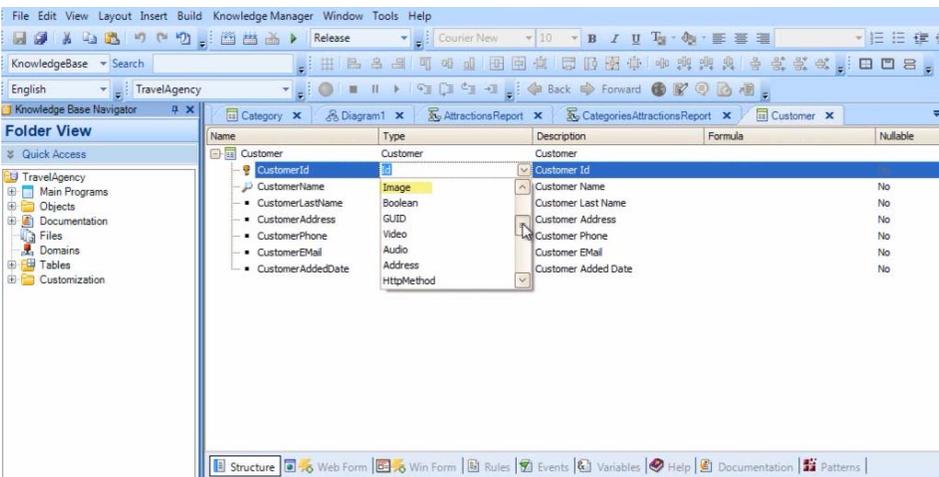
Name	Type	Description	Formula	Nullable
Customer	Customer	Customer		
CustomerId	Id	Customer Id		No
CustomerName	Name	Customer Name		No
CustomerLastName	Name	Customer Last Name		No
CustomerAddress	Address	Customer Address		No
CustomerPhone	Character(15)	Customer Phone		No
CustomerEMail	Character(50)	Customer EMail		No
CustomerAddedDate	Date	Customer Added Date		No

Hasta ahora hemos empleado siempre tipos de datos simples.

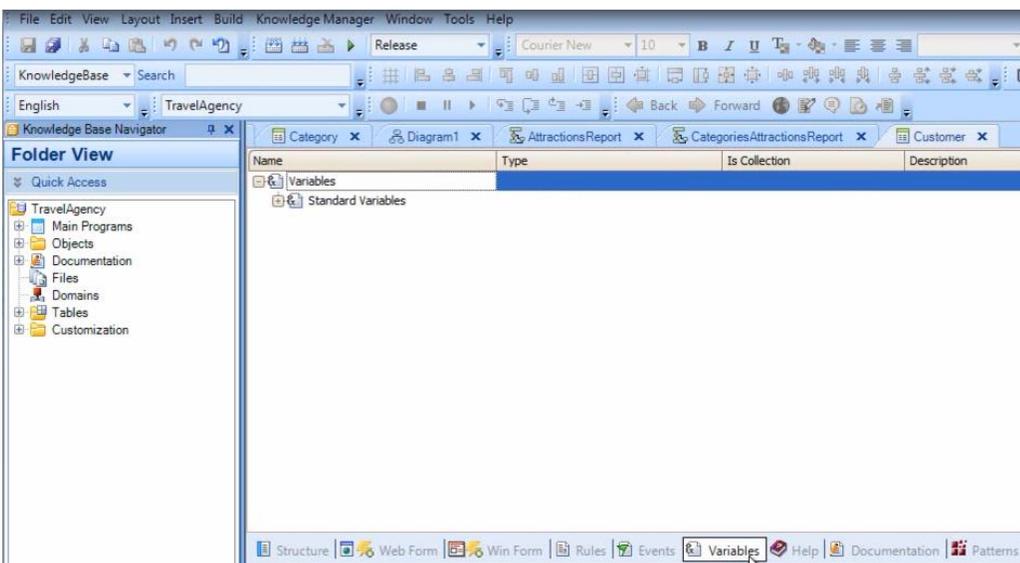
Hemos definido atributos y dominios de tipo Numeric, de tipo Character...



y también contamos con otros tipos de datos simples por ejemplo: Date, Image y otros



Además de atributos y dominios, también podemos definir **variables** de tipos de datos simples

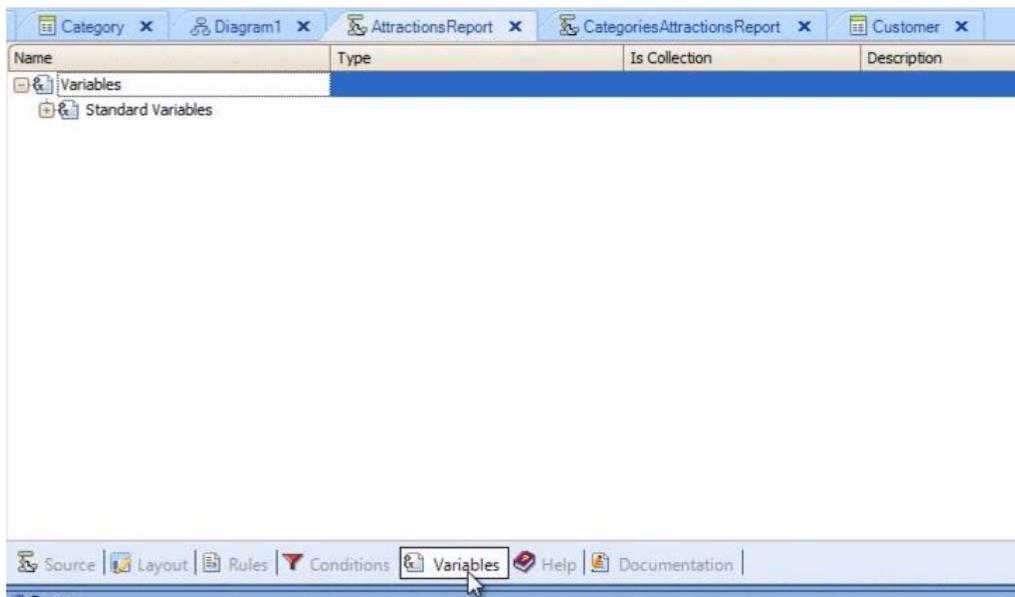


Una variable es una porción de memoria, a la cual le damos un nombre y nos sirve para guardar un dato en

forma temporal.

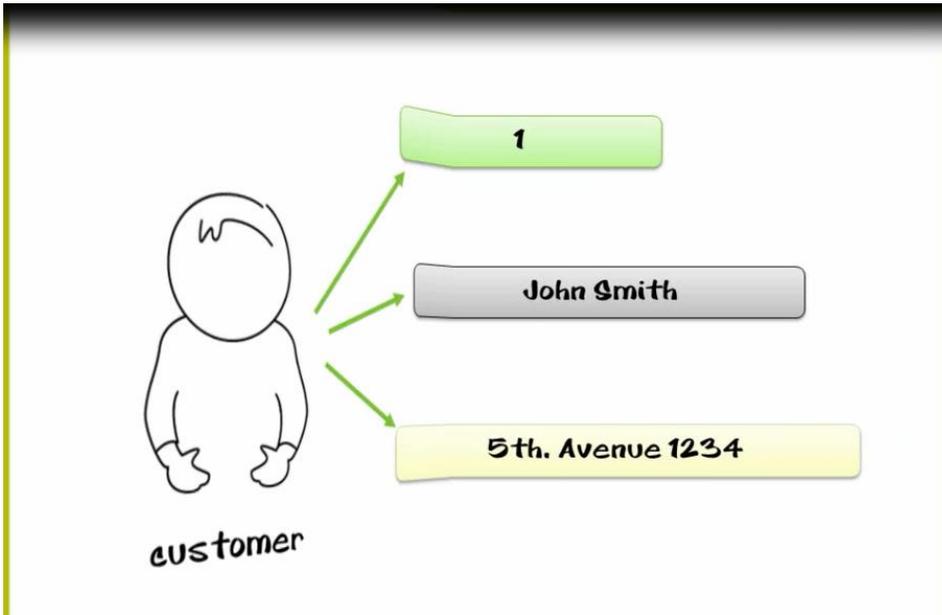


Como vemos, cada objeto tiene su sección de variables

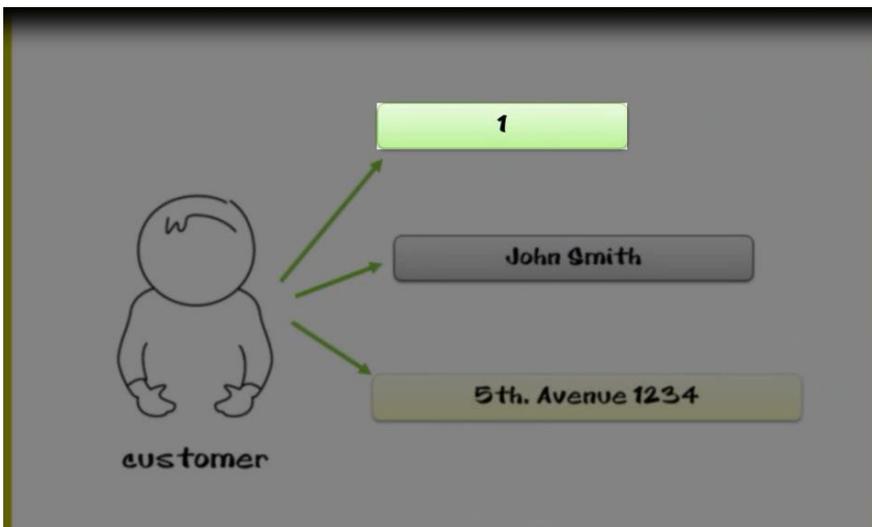


o sea que las variables que se definen en un objeto, son conocidas solamente en dicho objeto.

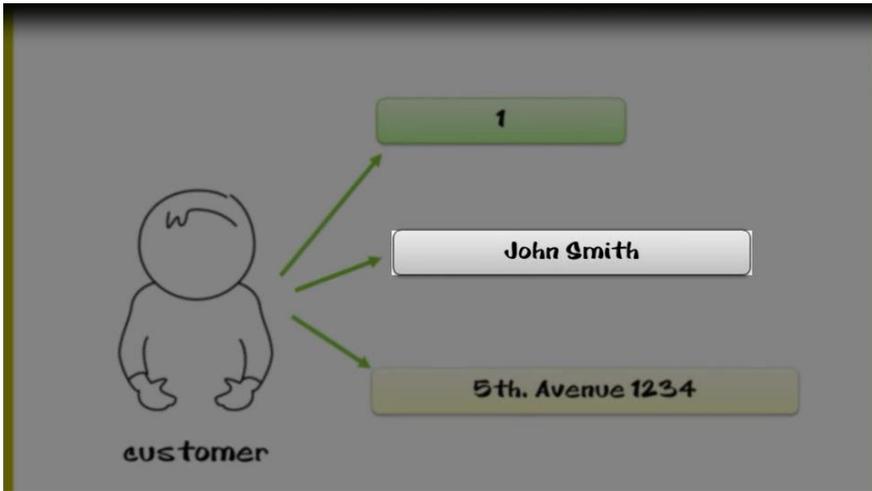
Ahora bien, si por ejemplo queremos guardar temporalmente los datos de 1 cliente en determinado objeto



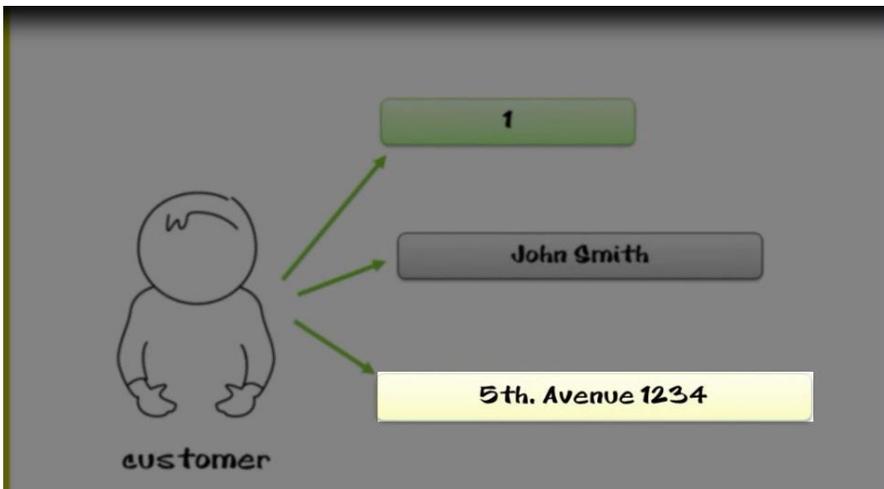
contamos con la opción de definir una variable individual para guardar el **id**



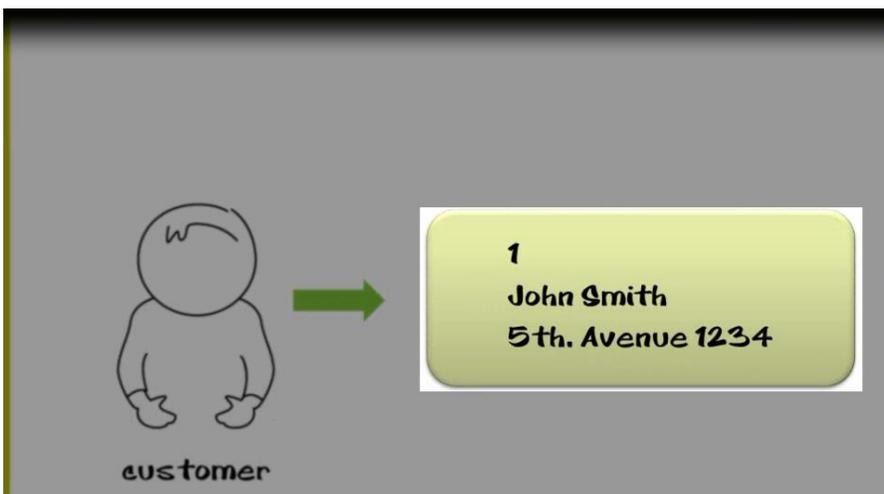
otra variable individual para guardar el **nombre**



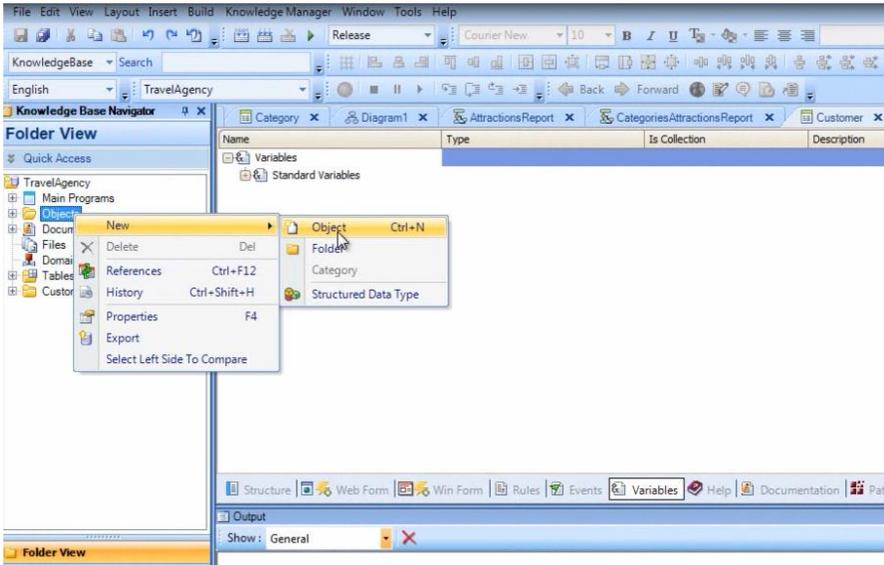
otra más para la **dirección**, etc.



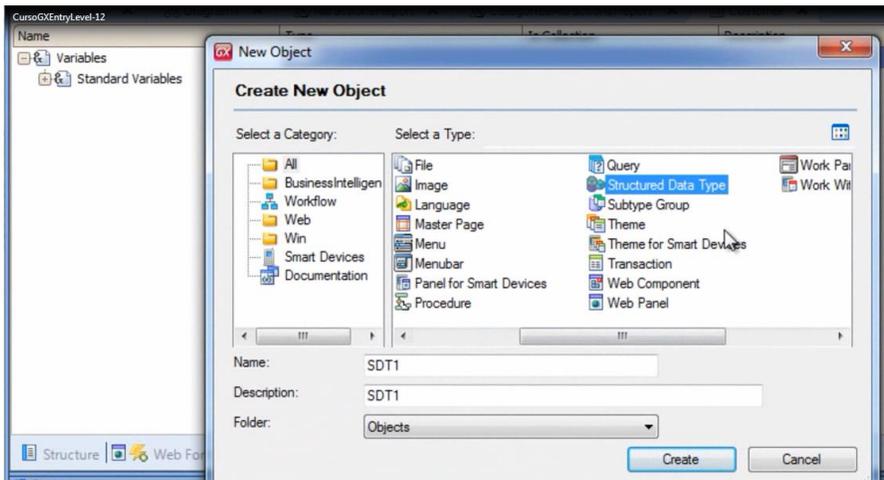
O sino, GeneXus nos permite también almacenar varios datos juntos en una sola variable



Para resolverlo de esta última forma, tenemos que definir un tipo de datos especial



llamado “tipo de datos compuesto o tipo de datos estructurado”



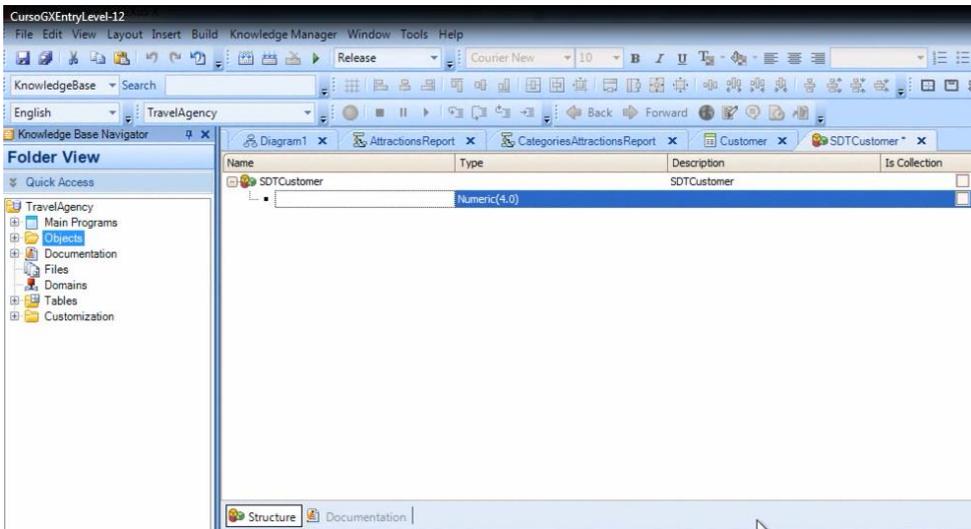
En inglés: “Structured Data Type” o SDT.

Le damos el nombre SDTCustomer



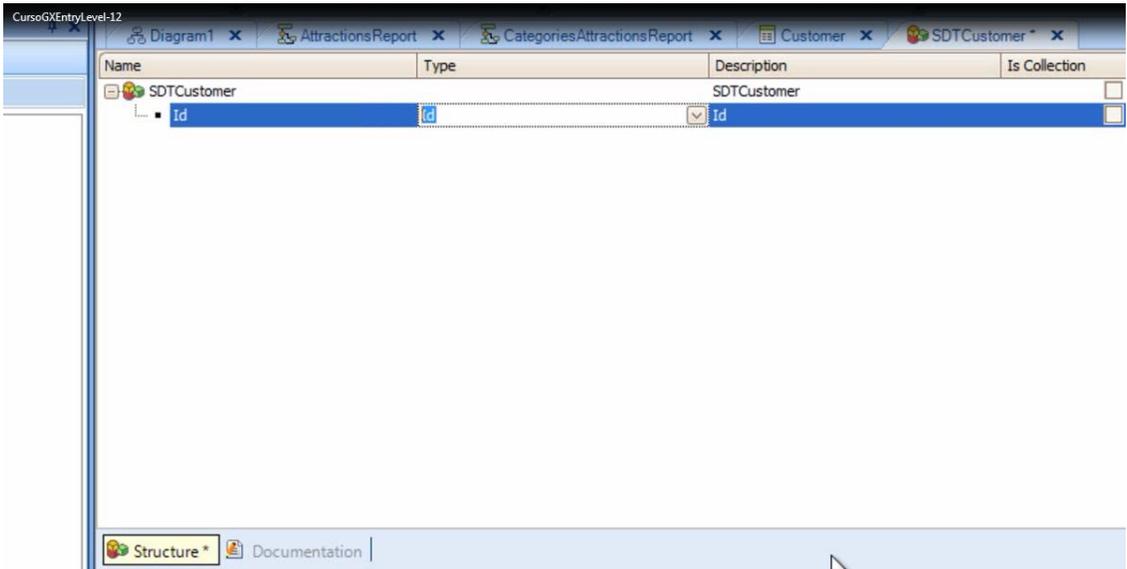
Create...

y vemos una estructura para definir, similar a la estructura de las transacciones

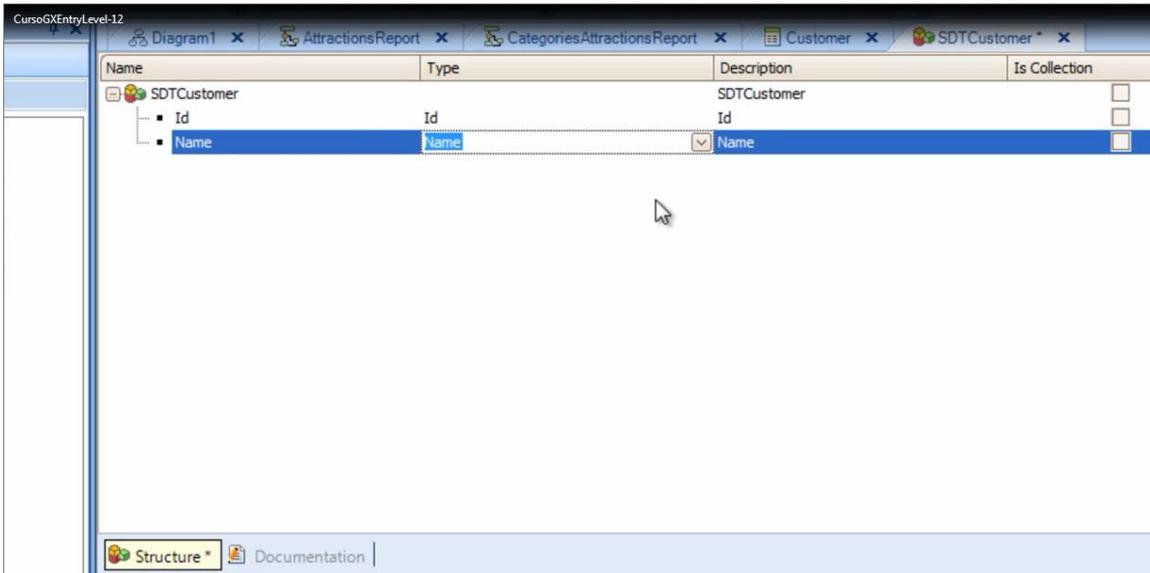


En ella, vamos a agregar cada miembro o nombre de dato que nos interesa guardar de 1 cliente y su tipo.

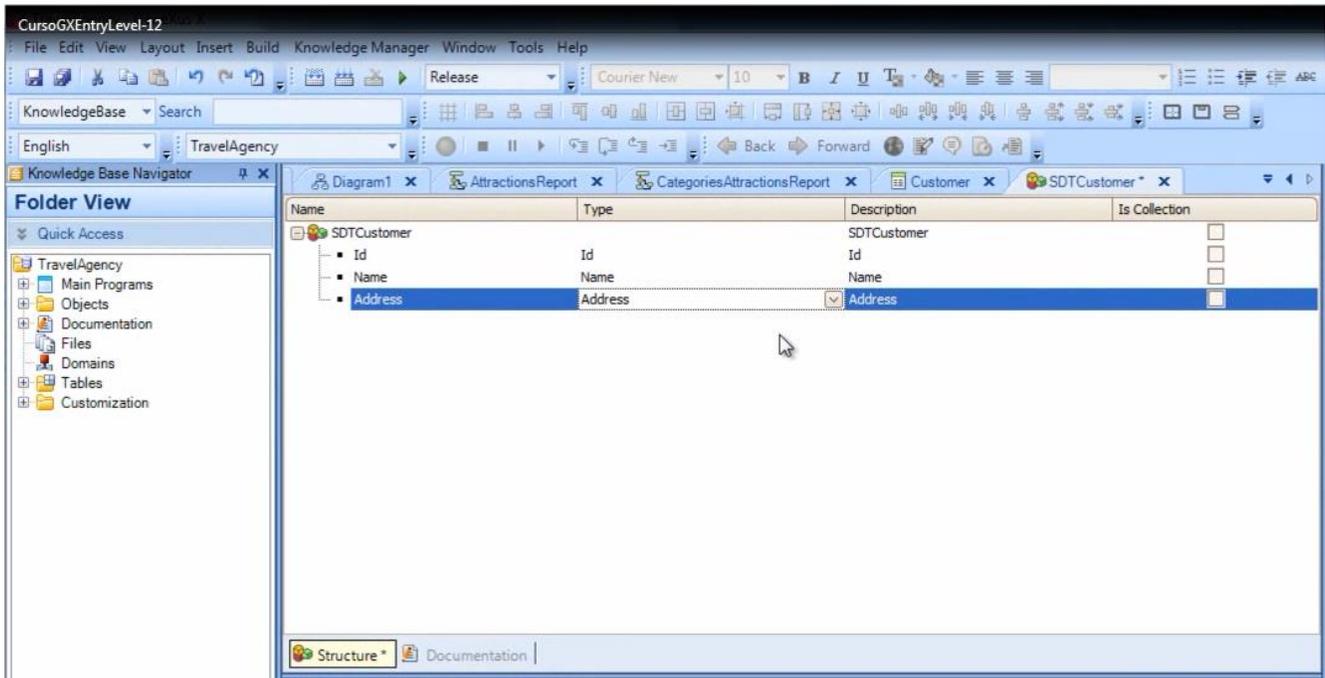
Agregamos un miembro de nombre Id y vemos que el tipo se asume por defecto del dominio Id, al igual que cuando incluimos al atributo CustomerId en la transacción Customer



Continuamos agregando a un miembro de nombre Name para almacenar el nombre del cliente



y un miembro Address para almacenar su dirección

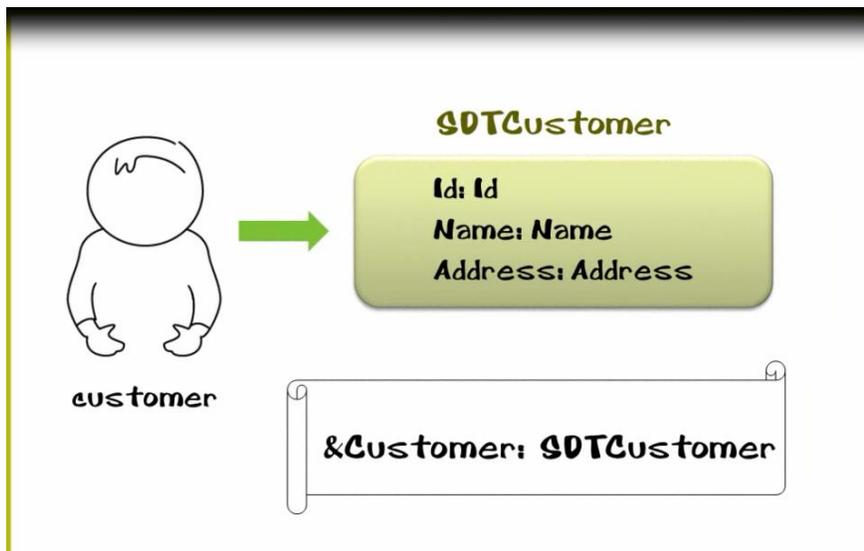


y salvamos.

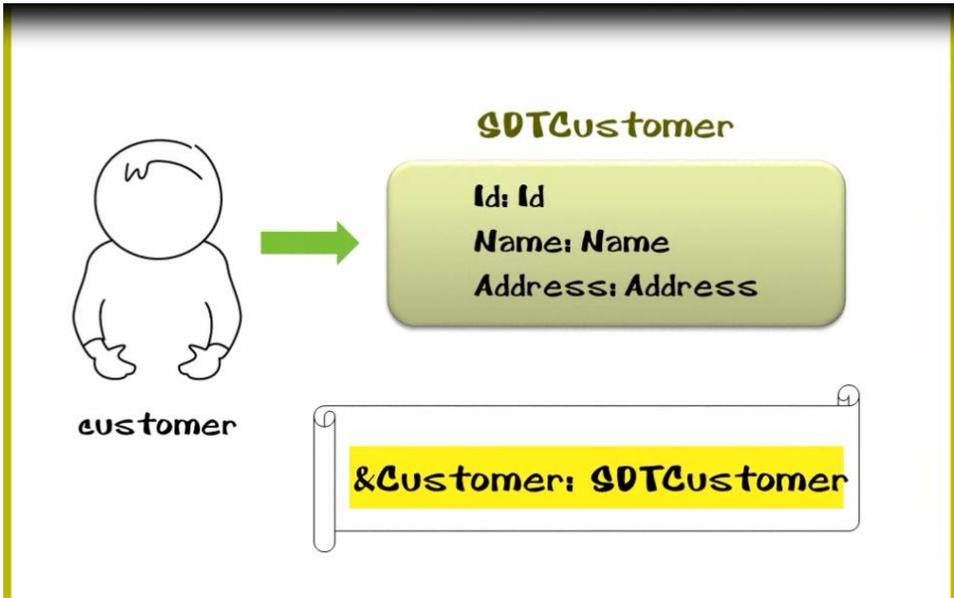
Esta definición que hicimos de un tipo de datos compuesto, podremos asignarla como tipo de datos de una **variable** que definamos en un objeto.

En cambio, no podemos usar un tipo de datos estructurado para definir un atributo, ya que éstos solamente pueden almacenar datos simples.

Por ejemplo

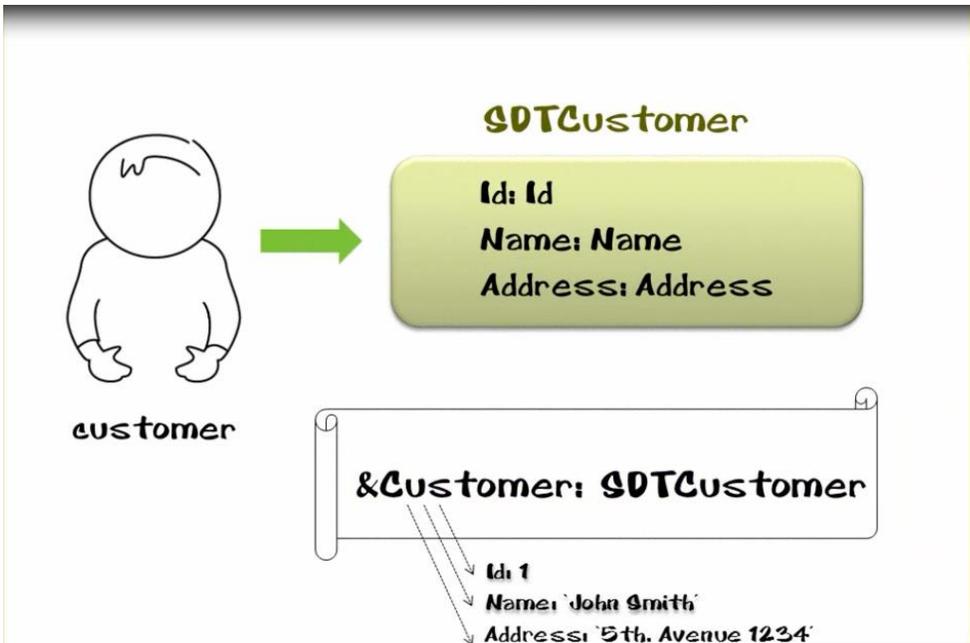


si queremos guardar los datos de un cliente en memoria, en cierto objeto para usarlos después, podríamos definir una variable

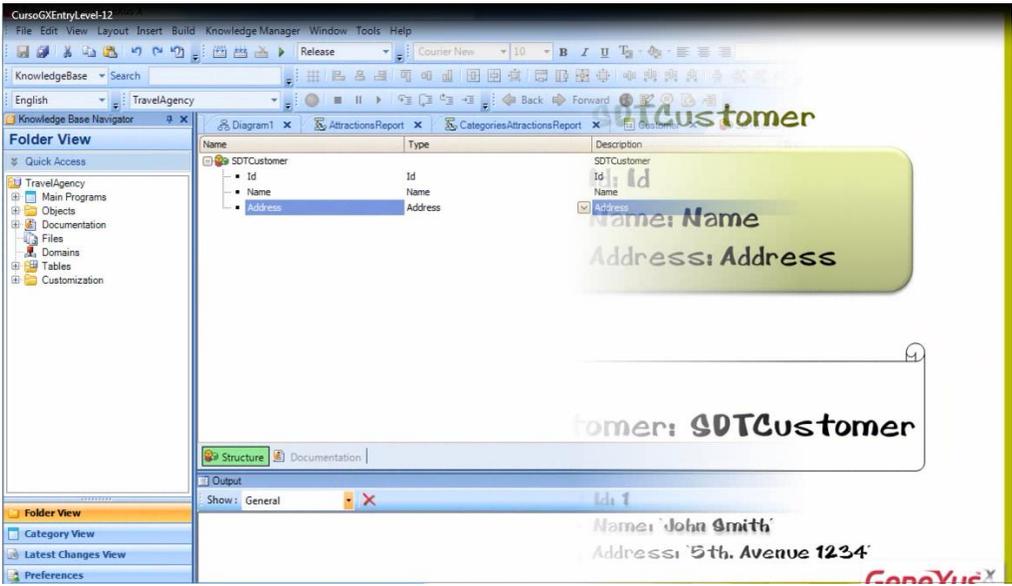


&Customer del tipo SDTCustomer

y cargaríamos la variable con los datos de un cliente

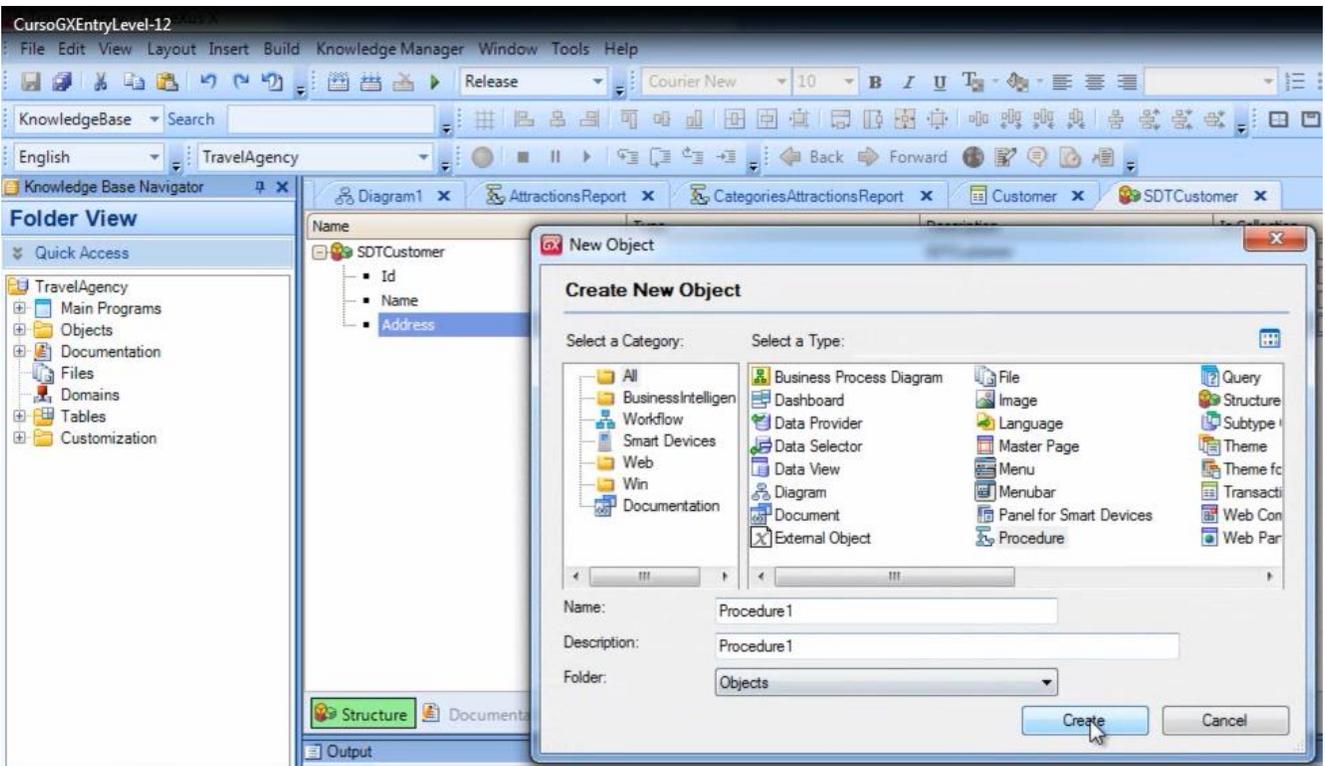


Veamos esto en GeneXus...



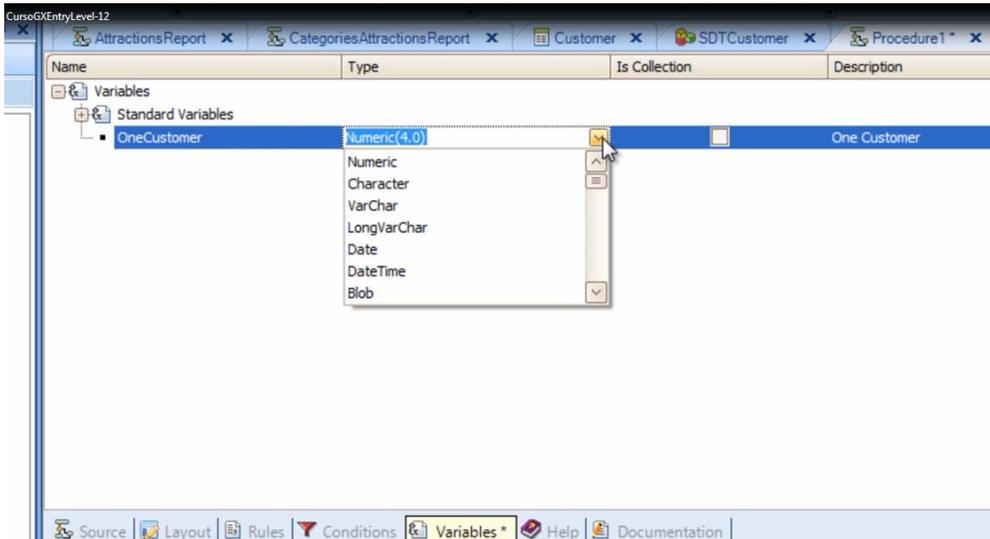
Podríamos usar variables en muchos objetos. En este caso vamos a hacerlo en un objeto procedimiento.

Vamos a crear un procedimiento, hacemos Control + N, le dejamos el nombre por defecto...y presionamos Create

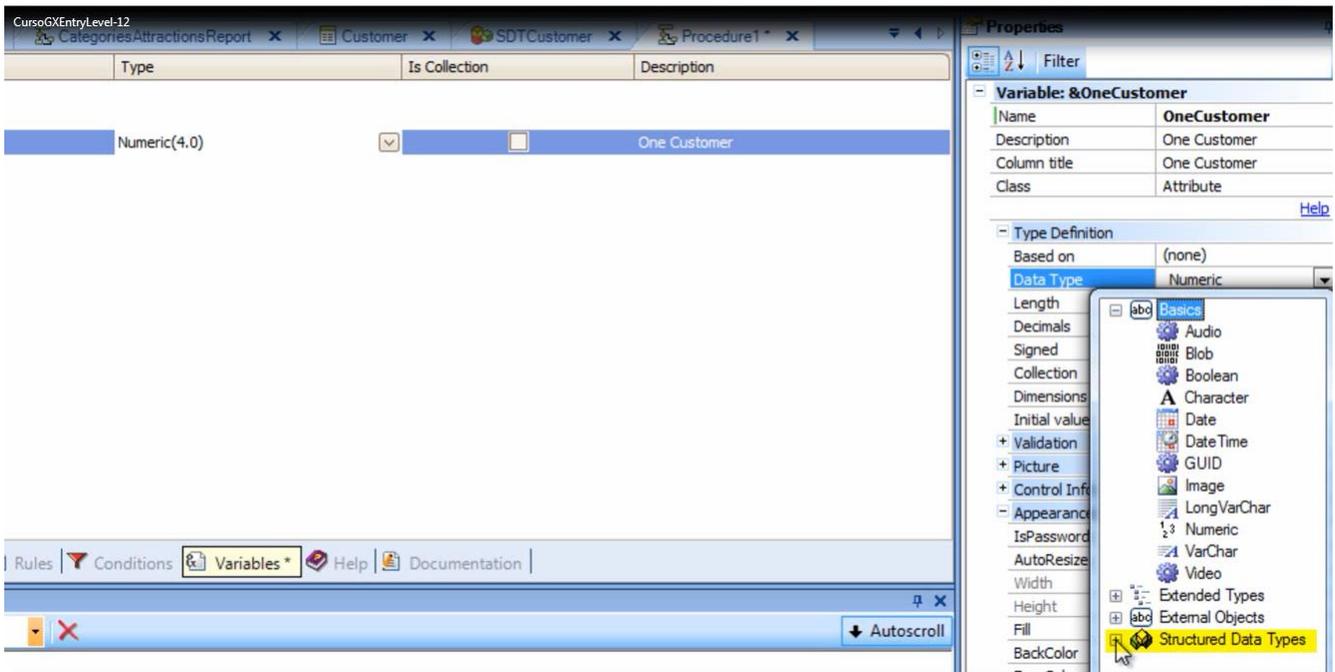


Ahora vamos a la sección Variables del procedimiento

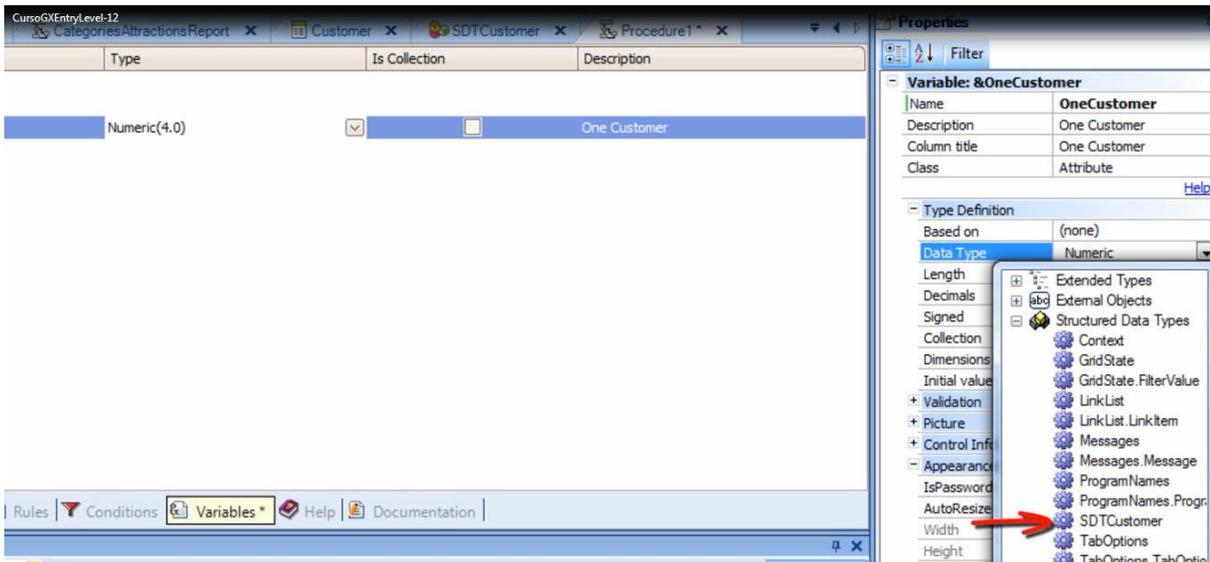




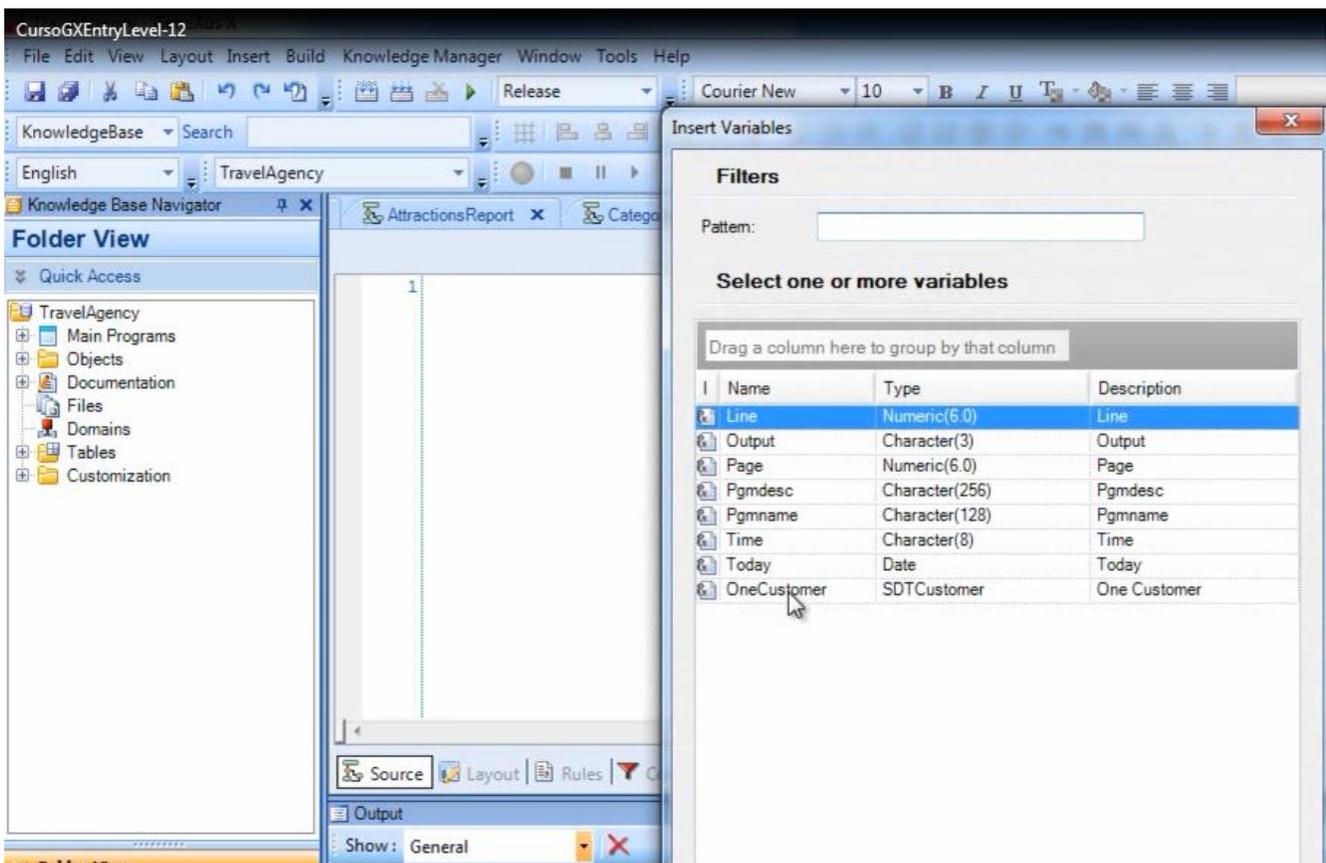
pero es más cómodo si vamos a la ventana de propiedades, hacemos clic en la propiedad DataType, presionamos la flecha y buscamos a los tipos de datos estructurados....



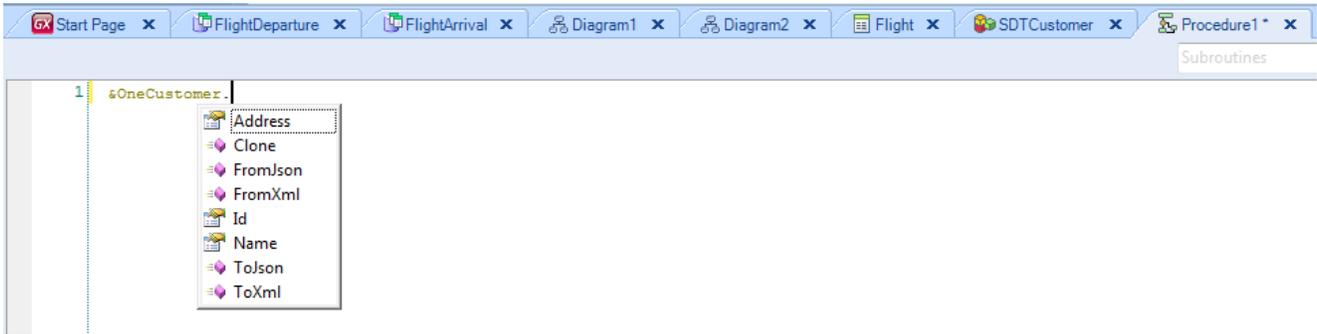
Hacemos clic en el signo de + y elegimos SDTCustomer



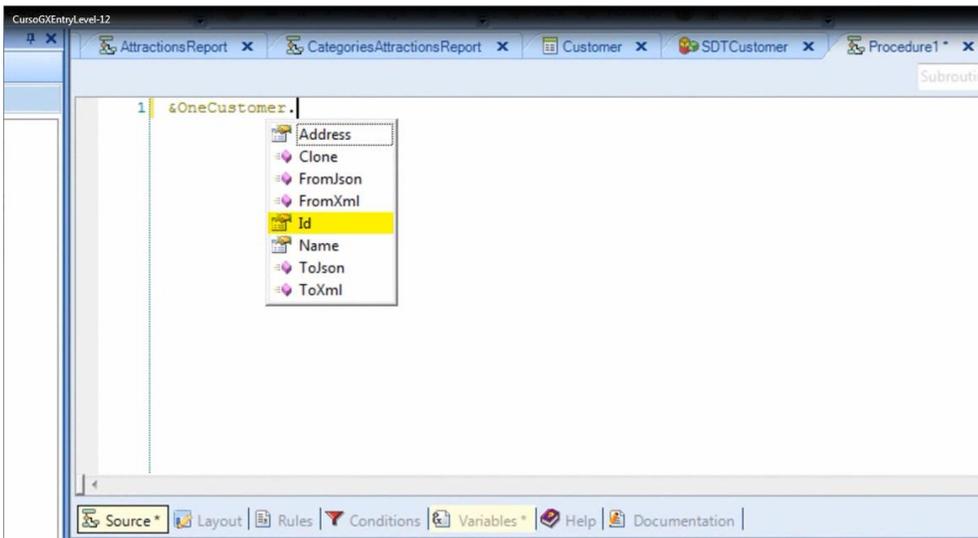
Ahora abrimos la sección source... en el menú Insert elegimos Variable...y escogemos a la variable OneCustomer:



Digitamos punto (".")



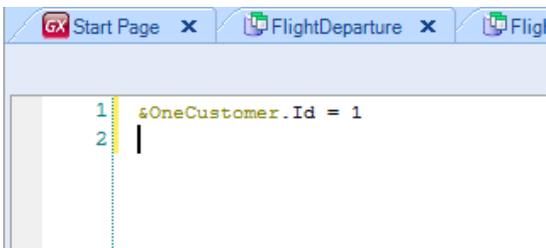
Y vemos que se abre una ventana en la cual vemos los miembros que creamos en el tipo de datos compuesto SDTCustomer



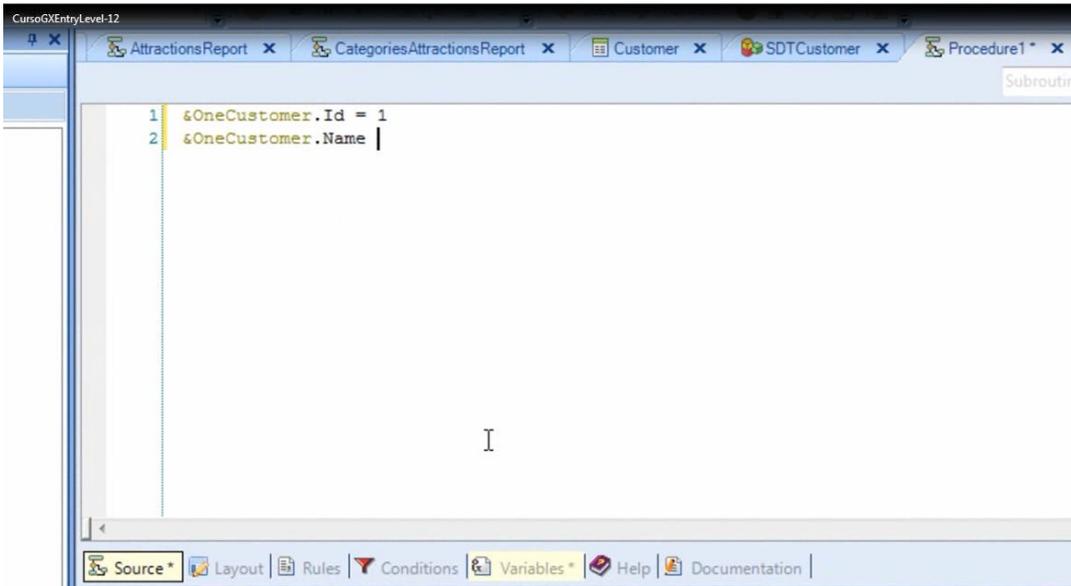
Address, Id, Name.

Elegimos Id y presionamos Enter.

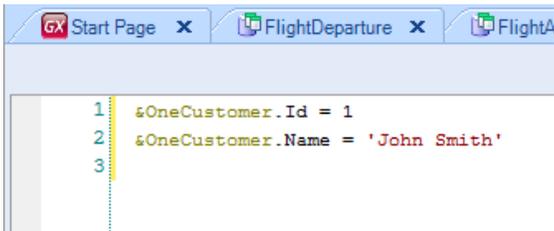
Ahora vamos a cargar el miembro Id de esta variable de tipo SDTCustomer con el identificador del cliente. Escribimos el signo de asignación (que es el signo de igual) y el número 1.



Continuemos cargando el nombre del cliente... insertamos en la 2da línea de código a la variable &OneCustomer nuevamente, digitamos punto y elegimos Name.

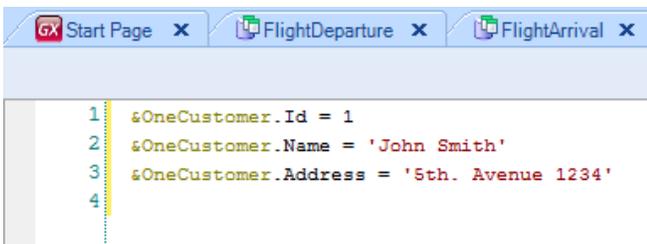


Le asignamos el nombre 'John Smith'



Y por último cargaremos la dirección.

Insertamos a la variable &OneCustomer, punto, Address y le asignamos: '5th. Avenue 1234'.



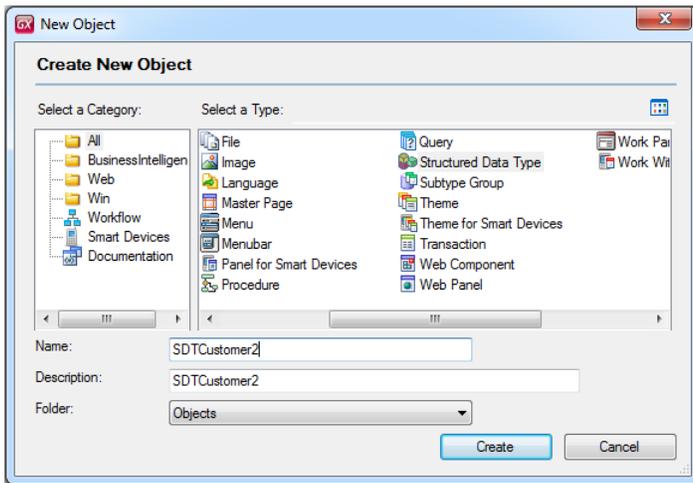
De esta manera hemos guardado en memoria, en la variable &OneCustomer ciertos datos específicos correspondientes a **un cliente**, y vamos a poder trabajar con los mismos, más adelante en el procedimiento, en alguna operación que los requiera.

Muchas veces nos resulta necesario guardar en memoria datos que están almacenados en la base de datos. En este caso, la estructura del tipo de datos estructurado que debemos crear, coincide con la estructura de atributos de la transacción que habíamos definido.

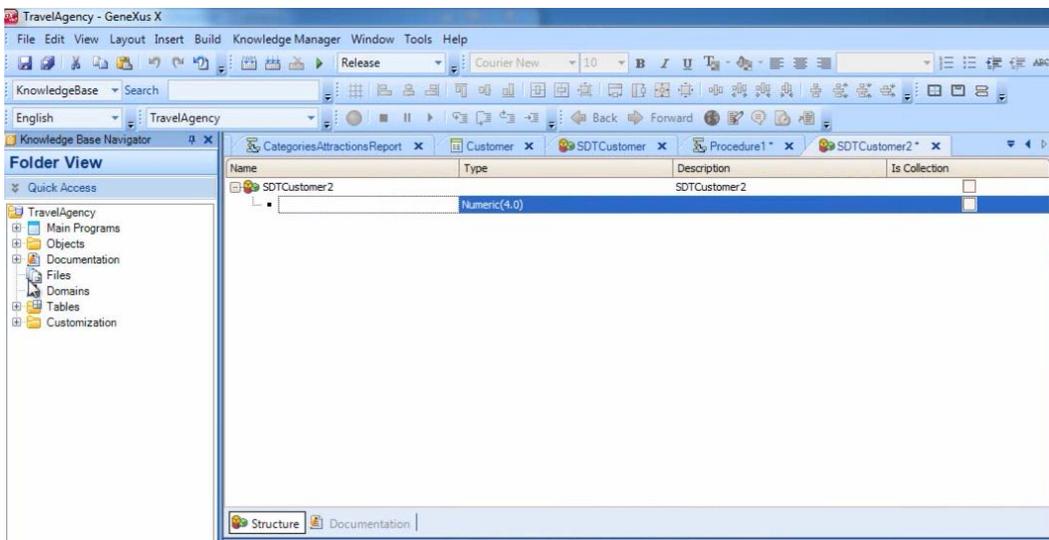
Veamos como GeneXus nos ayuda a resolver esto fácilmente ...

Creamos un nuevo tipo de datos estructurado... New Object...

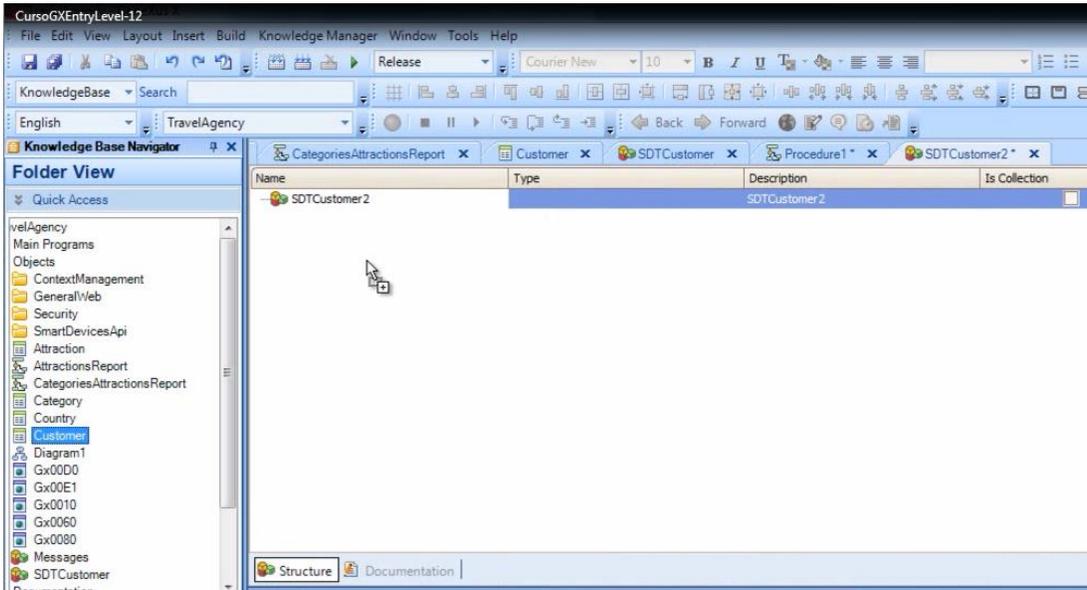
elegimos “Structured Data Type”



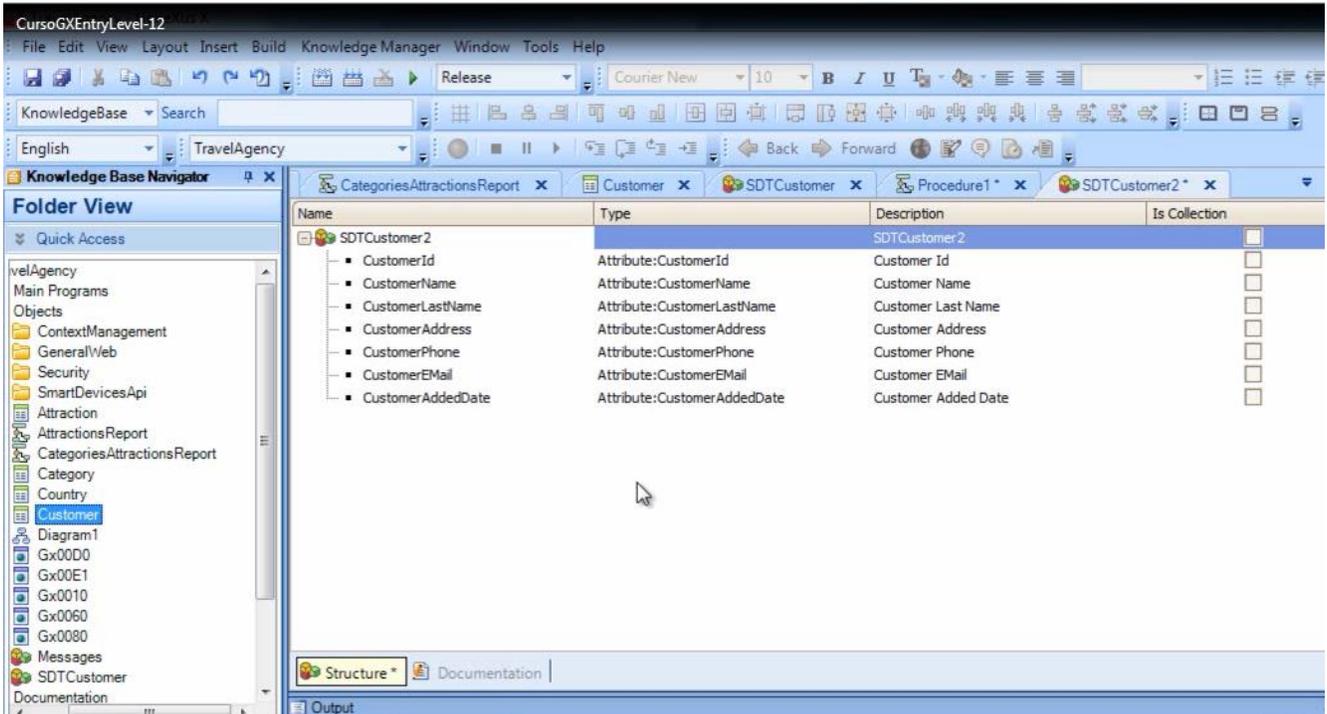
y le damos el nombre SDTCustomer2



Ahora, en lugar de empezar a definir los miembros del SDT uno a uno, vamos a la ventana Folder View Navigator, localizamos a la transacción Customer,



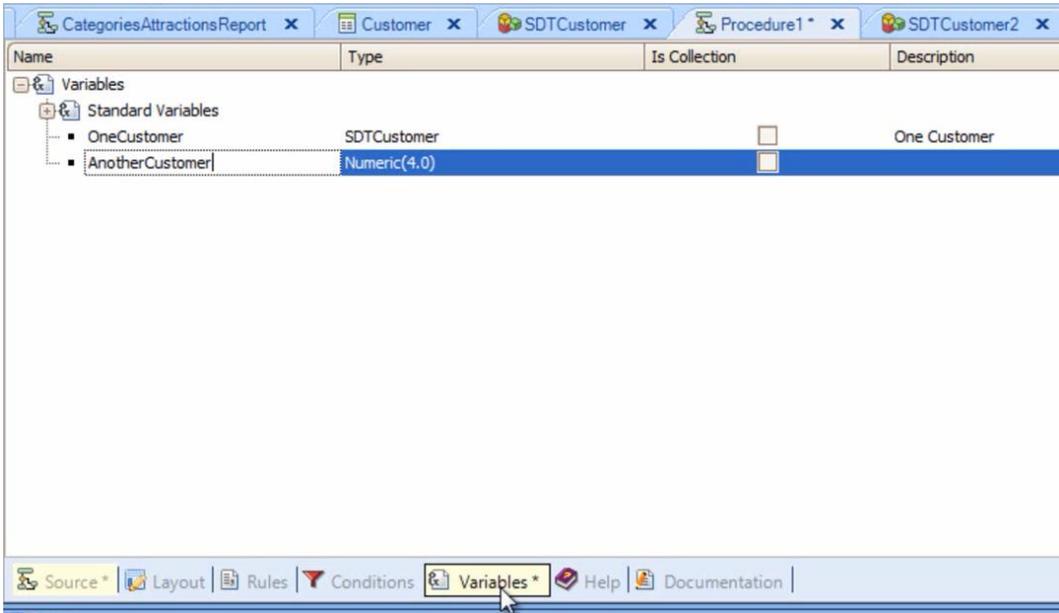
y arrastramos a la transacción sobre la estructura del SDT que estábamos definiendo y soltamos



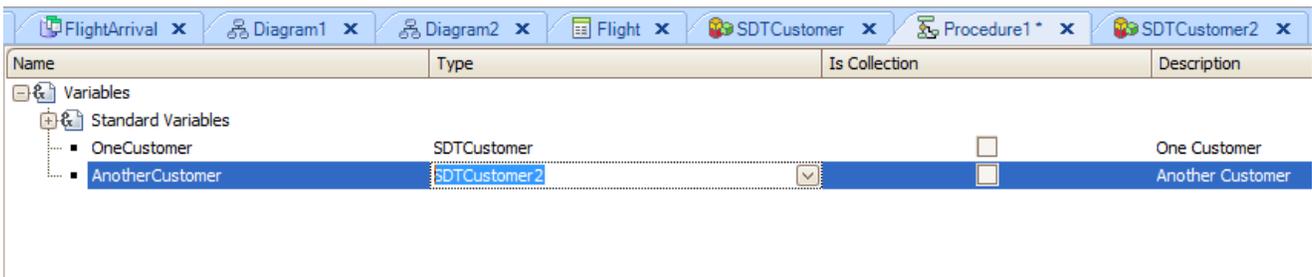
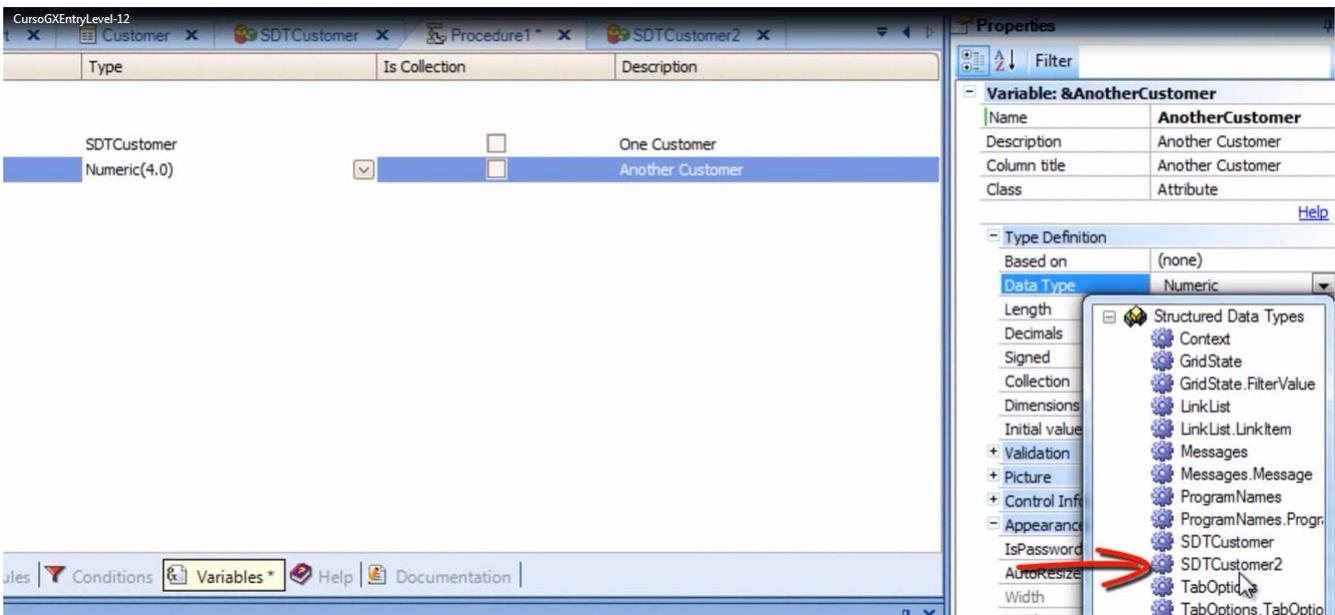
Vemos que se crearon automáticamente los miembros del SDTCustomer2, con los mismos nombres de los atributos de la transacción Customer....

Salvamos...

Ahora volvamos al procedimiento y definimos una variable de nombre &AnotherCostumer

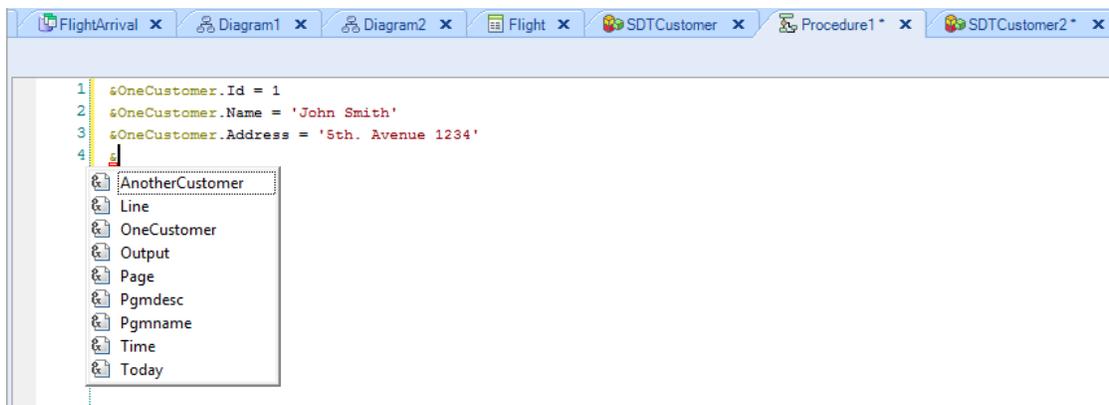


presionamos tabulador y en la ventana propiedades elegimos el tipo de datos SDTCustomer2



Volvamos a la sección Source e insertamos la variable que acabamos de definir.

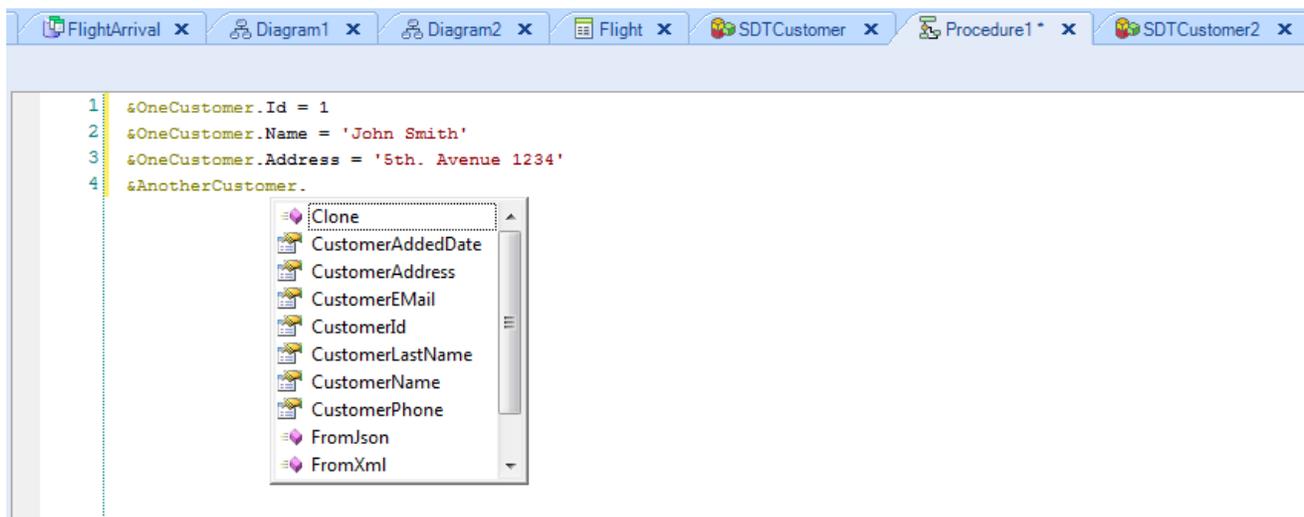
Podemos digitar el símbolo de ampersand (&)



y se nos ofrecerán todas la variables que tengamos definidas en el objeto.

Vemos que muchas de ellas no las creamos nosotros sino que las creó GeneXus.

Elegimos &AnotherCustomer, ahora digitamos punto (".")

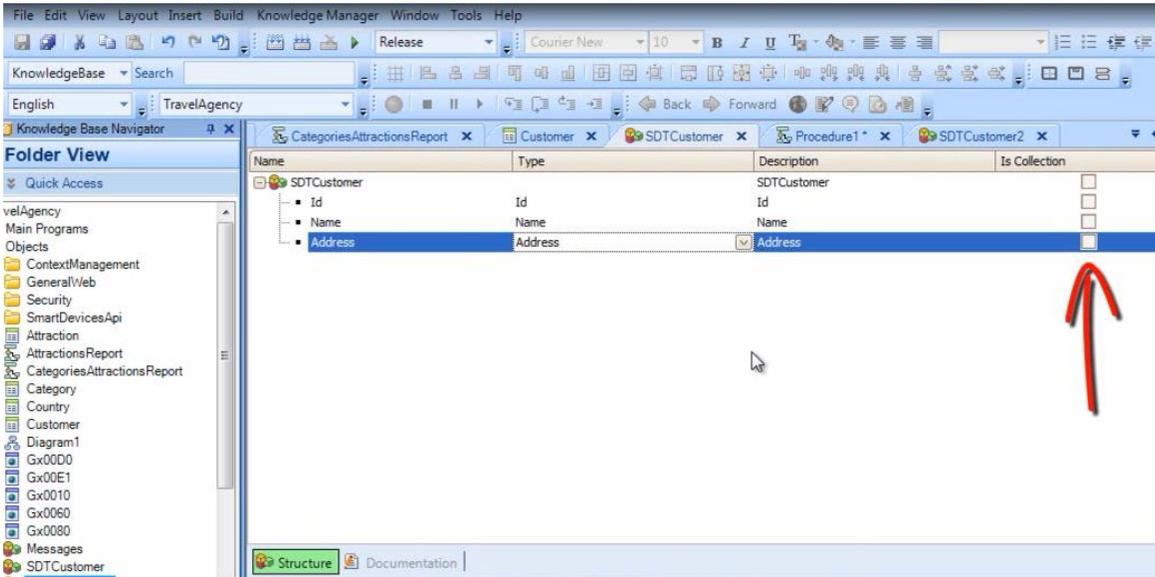


y vemos que aparecen todos los miembros del tipo de datos estructurado SDTCustomer2.

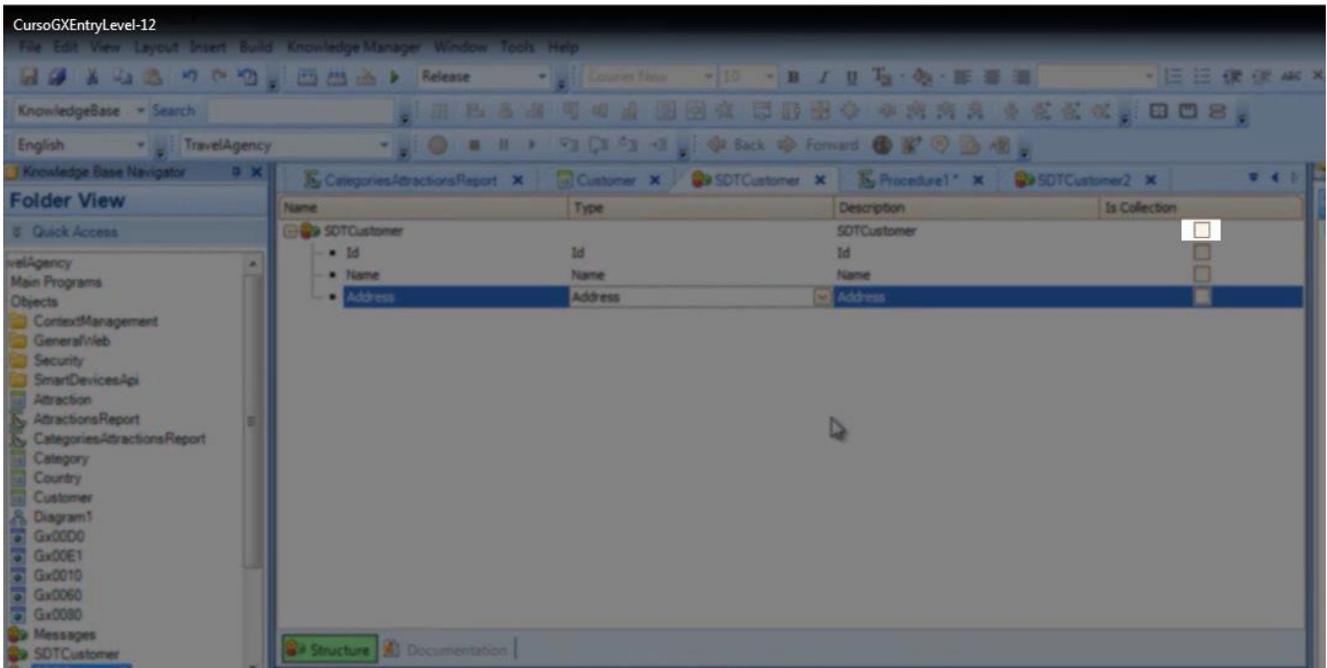
Habíamos dicho que nuestro objetivo, era cargar esta variable de tipo de datos estructurado, con datos que están almacenados en la base de datos. Veremos cómo realizar la carga, cuando estudiemos Data Providers.

Para terminar, si bien hasta ahora hemos mostrado el uso de un tipo de datos estructurado para guardar en memoria temporal los datos **de un cliente**, veamos que muy fácilmente podemos modificar su definición para guardar los datos **de muchos clientes**.

Volvamos a la definición del SDTCustomer

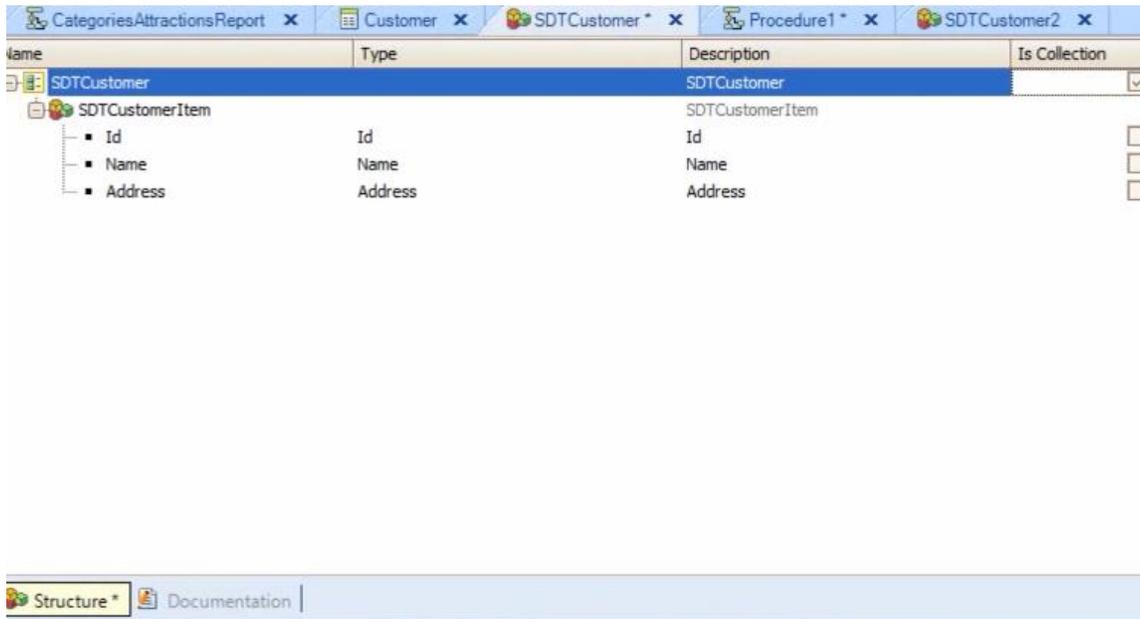


y observamos esta columna "Is Collection" que ofrece una casilla en la 1er línea

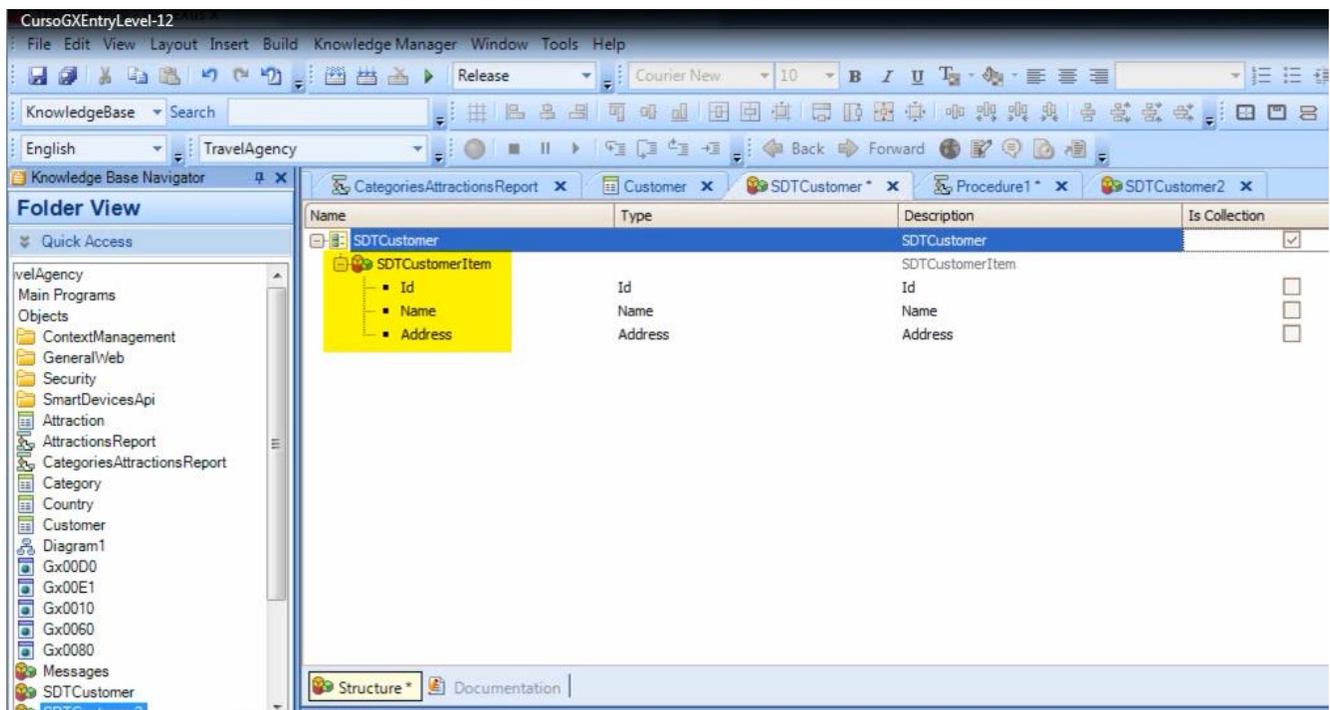


a la derecha del nombre SDTCustomer.

Si marcamos esta casilla



estaremos definiendo que el SDT almacenará una colección de elementos de la estructura definida



en lugar de un solo elemento.

Veremos más sobre esto cuando estudiemos Data Providers.