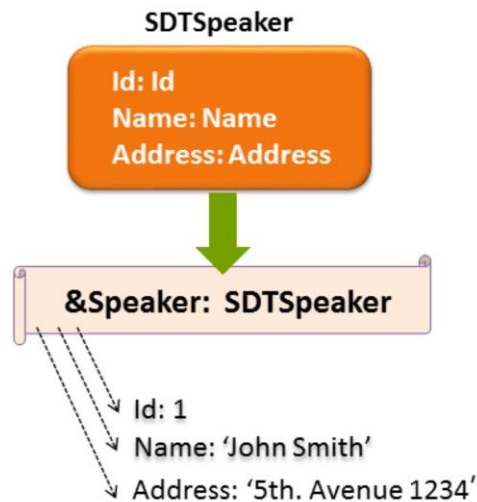


GeneXus X Evolution 3

Data Providers

En más de una oportunidad, necesitamos almacenar en memoria una **lista de elementos** que tienen el mismo tipo de información pero distintos valores guardados.

Generalidades



Supongamos que se necesita realizar operaciones con un grupo de oradores que cumplan cierto requisito, y eso puede implicar tener que cargar esta lista temporalmente en memoria.

Para resolver este tipo de requisito, es necesario crear una estructura en memoria **capaz de almacenar una colección de elementos**.

Ya hemos visto que los tipos de datos estructurados nos permiten definir estructuras que almacenan varios datos correspondientes a **un elemento**.

En esta representación gráfica entonces, estamos almacenando en memoria el identificador, el nombre y la dirección de **un** cliente.

SDTSpeaker

Id: Id
Name: Name
Address: Address



One speaker

SDTSpeakers

Id: 1
Name: John Smith
Address: 5th. Ave.

Id: 2
Name: Susan Brown
Address: 7th.Ave.

Id: 3
Name: Robert Hill
Address: 81th. St..

Id: 4
Name: Peter Jensen
Address: St,Paul Rd.

.....

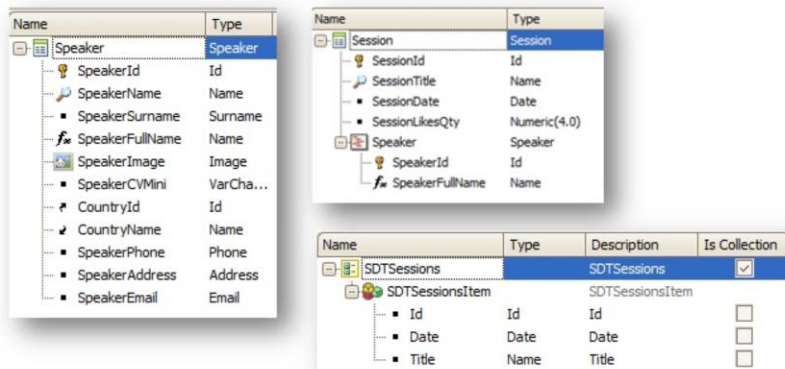


A collection of speakers

Para almacenar varios elementos con datos de oradores, es necesario definir un tipo de datos estructurado y marcar que se trata de una colección.

Ejemplo

Se necesita almacenar en memoria el conjunto de conferencias con más de dos oradores.



Veamos un ejemplo.

Se necesita almacenar en memoria el conjunto de conferencias que tienen más de dos oradores.

La estructura que es colección tiene los miembros agrupados en una subestructura denominada SDTSessionsitem. Esta subestructura se creó automáticamente cuando marcamos que se trata de una colección

Cada ítem almacenará los datos de un orador, y la colección al conjunto de oradores.

Collection structure



Para cargar los datos de la colección vamos a utilizar un objeto GeneXus de tipo **Data Provider**.

Este objeto nos permite cargar una estructura de datos, por ejemplo a partir de datos de la base de datos, y nos devuelve dicha estructura ya cargada.

Ejemplo (continuación)

Arrastrar el SDT sobre el source del Data Provider

```
SDTSessions
{
  SDTSessionsItem
  {
    id = /*id value*/
    Date = /*Date value*/
    Title = /*Title value*/
  }
}
```

Data Provider: DPSessions	
Name	DPSessions
Description	DPSessions
Expose as Web Service	False
Module/Folder	Root Module
Qualified Name	DPSessions
Object Visibility	Public
Output	
Infer Structure	No
Output	SDTSessions
Collection	False

Creamos un objeto Data Provider en GeneXus y lo nombramos: DPSpeakers

GeneXus nos posiciona en la sección Source del Data Provider. Aquí es donde vamos a declarar cómo queremos que se carguen los datos en la colección que queremos devolver. Observemos qué fácil resulta declarar la carga: Vamos a la ventana del Knowledge Base Navigator, ubicamos al tipo de datos estructurado SDTSessions y lo arrastramos hacia el source.

Por realizar el arrastre, GeneXus escribe automáticamente varias líneas de texto.

Si abrimos las propiedades del DataProvider, observamos que GeneXus asignó el nombre de la colección SDTSpeakers a la propiedad Output. **Esto significa que el DataProvider devolverá una colección del tipo de datos estructurado SDTCountries, cargada con datos.**

Como el SDTSpeakers ya es una colección, no es necesario configurar la propiedad Collection con valor True. Esto lo haríamos, si quisiéramos que el DataProvider nos devuelva una colección a partir de un tipo de datos estructurado simple.

Ejemplo (continuación)

```
SDTSessions
{
  SDTSessionsItem
  {
    id = /*id value*/
    Date = /*Date value*/
    Title = /*Title value*/
  }
}
```

Nombre del tipo de datos estructurado.

Subestructura del ítem de la colección.

```
SDTSessions
{
  SDTSessionsItem
  {
    id = /*id value*/
    Date = /*Date value*/
    Title = /*Title value*/
  }
}
```

Estudiamos ahora qué fue lo que escribió GeneXus en el source.

Reconocemos el nombre del tipo de datos estructurado SDTSessions que es una colección. Y después entre llaves está la subestructura del ítem de la colección.

Ejemplo (continuación)

Name	Type	Description	Is Collection
SDTSessions		SDTSessions	<input checked="" type="checkbox"/>
SDTSessionsItem		SDTSessionsItem	
▪ Id	Id	Id	<input type="checkbox"/>
▪ Date	Date	Date	<input type="checkbox"/>
▪ Title	Name	Title	<input type="checkbox"/>

```
SDTSessions
{
  SDTSessionsItem
  {
    id = /*id value*/
    Date = /*Date value*/
    Title = /*Title value*/
  }
}
```

Comparemos esto con la estructura del SDT:

Vemos que GeneXus representó en forma de texto la estructura del SDTSessions. Y nos dejó prontos los miembros Id, Date y Title de la subestructura SDTSessionsitem para cargarles su valor.

Ejemplo (continuación)

```
SDTSessions from Session
where count(Speakerid) > 2
{
  SDTSessionsItem
  {
    id = /*id value*/
    Date = /*Date value*/
    Title = /*Title value*/
  }
}
```

Nombre de la transacción cuya tabla física deseamos que recorra el Data Provider

Condición que deben cumplir los datos a cargarse en la colección.

Se indican los atributos o cálculos con los cuales se cargarán los elementos de la colección:

```
SDTSessions from Session
where count(Speakerid) > 2
{
  SDTSessionsItem
  {
    id = SessionId
    Date = SessionDate
    Title = SessionTitle
  }
}
```

Como vamos a cargar esta colección a partir del contenido de la tabla SESSION, debemos indicarle al Data Provider que deberá recorrer dicha tabla. Para ello utilizamos la cláusula From y el nombre de la transacción cuya tabla base queremos recorrer.

En nuestro caso: From Session

Si la transacción tuviera más de un nivel, para especificar un nivel determinado, asociado a cierta tabla base que queremos navegar, tendríamos que escribir: **nombre de la transacción, punto, nombre del nivel**.

Pero no queremos cargar todas las conferencias sino solamente aquellas que tengan más de dos oradores, así que lo indicamos utilizando la cláusula Where al igual que en un For each.

Luego indicamos que al elemento Id lo cargamos con el valor de SessionId, al miembro Date con el valor del atributo SessionDate, y al miembro Title cargamos con SessionTitle.

Ejemplo (continuación)

Name	Type
Speaker	Speaker
SpeakerId	Id
SpeakerName	Name
SpeakerSurname	Surname
SpeakerFullName	Name
SpeakerImage	Image
SpeakerCVMini	VarCha...
CountryId	Id
CountryName	Name
SpeakerPhone	Phone
SpeakerAddress	Address
SpeakerEmail	Email

Name	Type
Session	Session
SessionId	Id
SessionTitle	Name
SessionDate	Date
SessionLikesQty	Numeric(4,0)
Speaker	Speaker
SpeakerId	Id
SpeakerFullName	Name

```
SDTSessions from Session
where count(SpeakerId) > 2
{
    SDTSessionsItem
    {
        id = SessionId
        Date = SessionDate
        Title = SessionTitle
    }
}
```

La tabla base del Data Provider es:
SESSION

De modo que lo que hemos hecho simplemente ha sido: **declarar 1 tabla a ser navegada por el Data Provider, y para cada registro accedido, hemos indicado los valores que deseamos asignar a un ítem nuevo en la colección de países.**

Dado que el Data Provider recorre la tabla SESSION, solemos decir que **la tabla base del Data Provider, es SESSION:**

El resultado final será que habrán quedado almacenados en la colección en memoria, los datos de todas las conferencias de la base de datos, que cumplen la condición de tener más de dos oradores.

GeneXus™