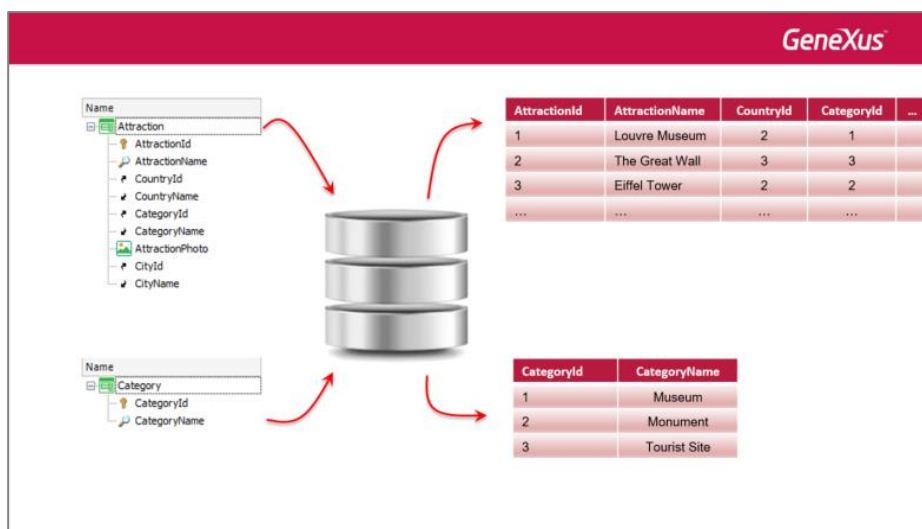


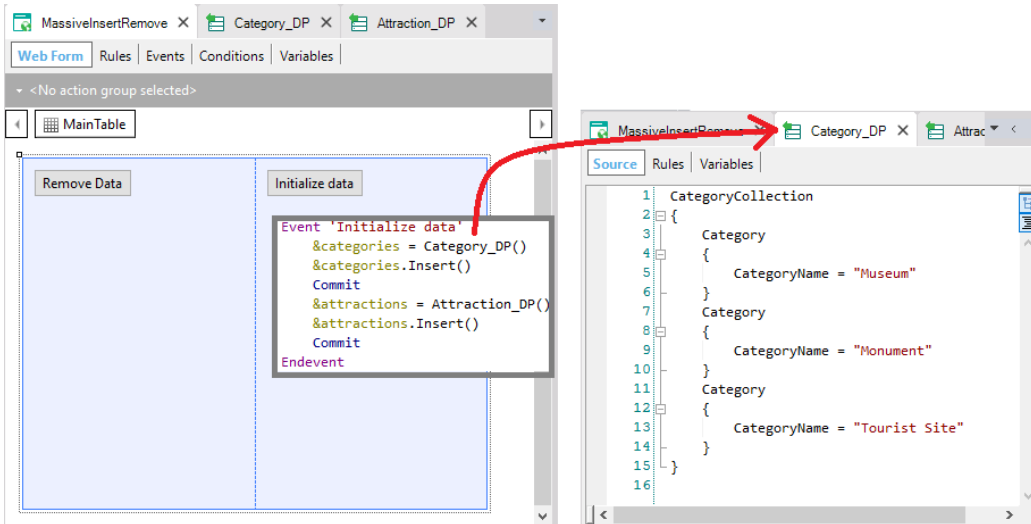
Poblar de datos las tablas desde la propia transacción



Cuando creamos una transacción, por defecto GeneXus creará tablas asociadas para almacenar la información que ingresamos a través de su pantalla.

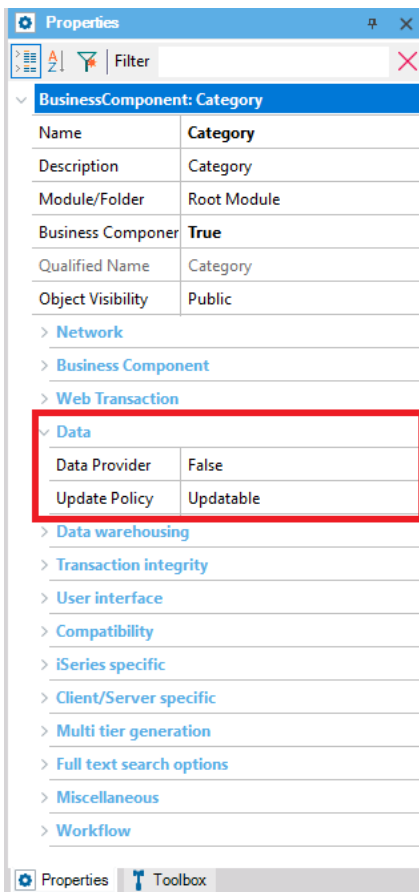


En el video anterior vimos que estas tablas podían ser inicializadas utilizando el Business Component asociado a la transacción, a través de una variable colección cargada mediante un Data Provider.



Pero GeneXus ya nos ofrece una solución para inicializar los datos correspondientes a una transacción, sin que tengamos que hacer todos los pasos anteriores manualmente (obtener el business component, crear el data provider, la variable colección, invocar el data provider, hacer el Insert).

Para ello la transacción cuenta con una propiedad, bajo el grupo Data, llamada **Data Provider**. Veámoslo con la transacción Category. Por defecto está en False:



... pero la pasaremos a True.

▼ Data

Data Provider	True
Used to	Populate data
Update Policy	Updatable

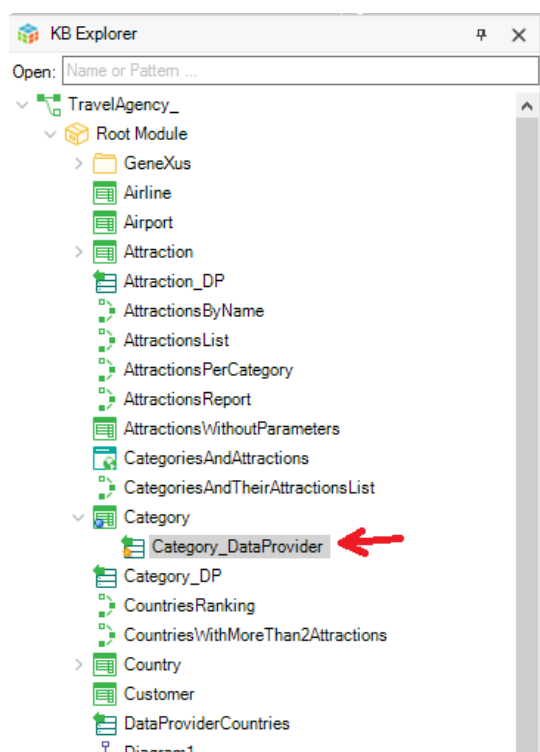
Con esto le estamos diciendo que existirá un Data Provider asociado. Y en esta nueva propiedad:

▼ Data

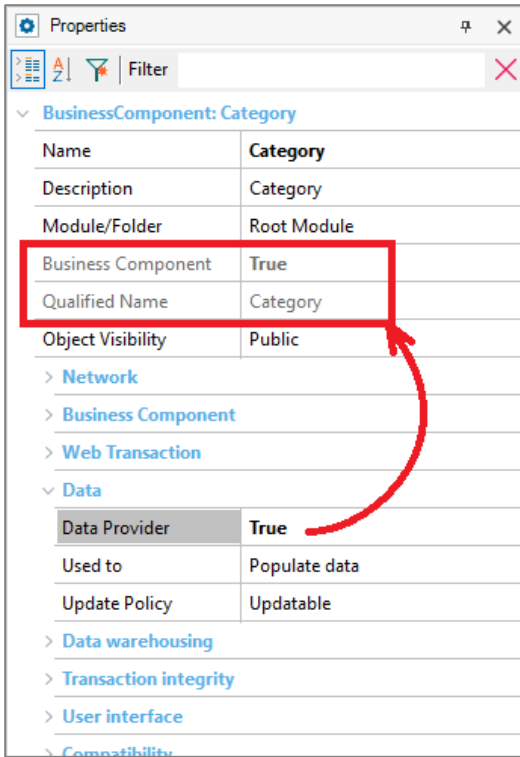
Data Provider	True
Used to	Populate data
Update Policy	Updatable

... le informamos que lo usaremos para Inicializar los datos de la tabla.

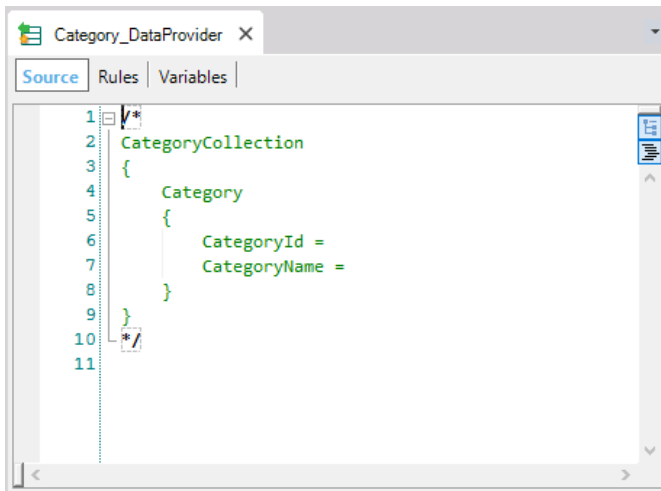
Así, vemos al grabar que se creó un objeto del tipo Data Provider, al que llamó Category_DataProvider.



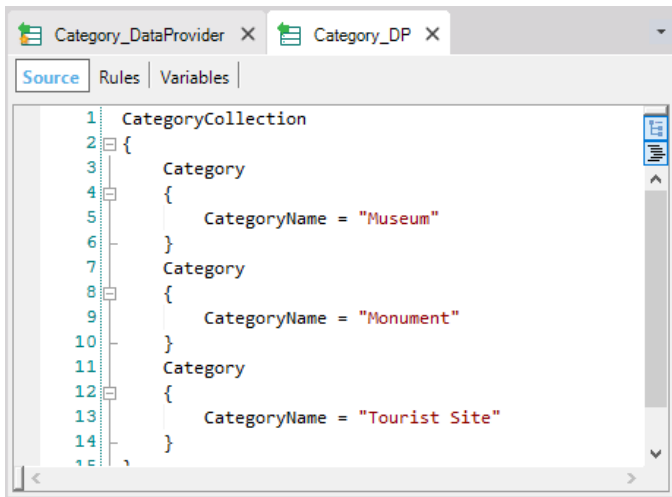
Además si no hubiéramos tenido en True la propiedad Business Component de la transacción, la hubiera puesto en True, haciendo que se cree entonces el business component asociado a la transacción.



Si abrimos el Data Provider podemos ver que ya nos ofrece el código para que simplemente completemos los datos de las categorías:

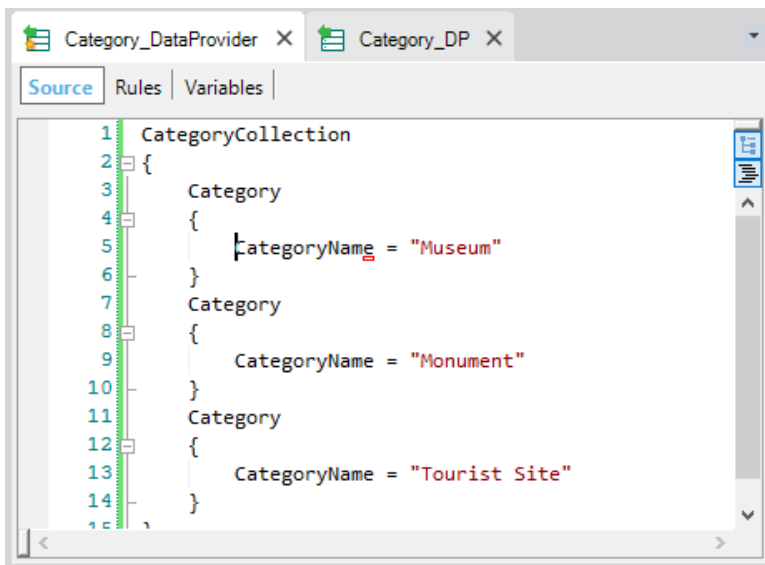


Este código es prácticamente un calco del que hicimos antes manualmente:



```
1 CategoryCollection
2 {
3   Category
4   {
5     CategoryName = "Museum"
6   }
7   Category
8   {
9     CategoryName = "Monument"
10  }
11  Category
12  {
13    CategoryName = "Tourist Site"
14  }
15 }
```

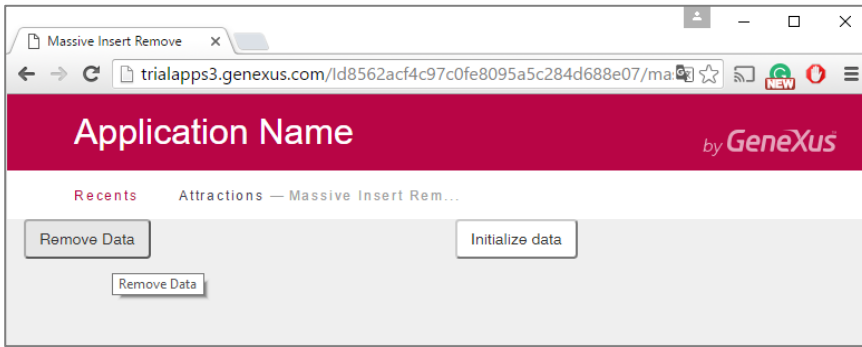
... por lo que copiamos el código que habíamos escrito hacia el nuevo data provider



```
1 CategoryCollection
2 {
3   Category
4   {
5     CategoryName = "Museum"
6   }
7   Category
8   {
9     CategoryName = "Monument"
10  }
11  Category
12  {
13    CategoryName = "Tourist Site"
14  }
15 }
```

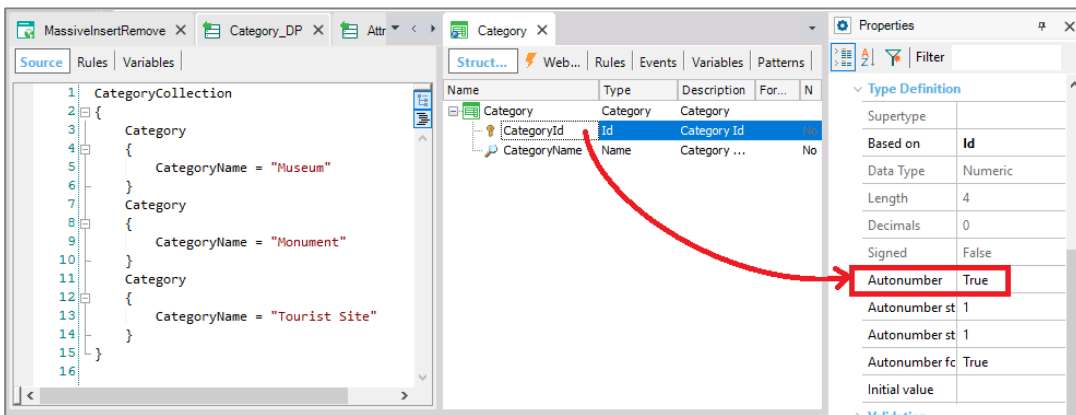
Con esto ya hemos definido cómo el Data Provider asignará valores a las nuevas categorías que se crearán. Lo que no hemos configurado aún es el momento en el que se invocará a este Data Provider para que realice la tarea. Si lo pensamos, el momento adecuado será cuando se crea la tabla en la base de datos. Y ese será el momento elegido por defecto por GeneXus para ejecutarlo.

Pero en nuestro caso la tabla ya está creada y no tendrá datos porque los vamos a eliminar. Presionamos Remove Data y con eso se eliminan los datos de Category y de Attraction.



Sin embargo, como acabamos de habilitar la inicialización de los datos de la transacción, entonces en la próxima ejecución GeneXus va a proceder a ejecutar el Data Provider.

Observemos que si la tabla ya tuviera datos en ese momento, debido a que tenemos el identificador como autonumber:



... lo que hará será agregar nuevos registros. No es nuestro caso, ya que tuvimos la precaución de vaciar la tabla antes. Si no fuera autonumber el identificador, y existieran registros con los valores que estamos agregando en el Data Provider, la inserción fallará.

Lo mismo haríamos con la transacción Attraction:

Properties

Filter

BusinessComponent: Attraction

Name	Attraction
Description	Attraction
Module/Folder	Root Module
Business Component	True
Qualified Name	Attraction
Object Visibility	Public

> Network

> Business Component

> Web Transaction

Data

Data Provider	True
Used to	Populate data
Update Policy	Updatable

> Data warehousing

> Transaction integrity

TravelAgency_

- Root Module
 - GeneXus
 - Airline
 - Airport
 - Attraction
 - WorkWithAttraction
 - Attraction_DataProvider
 - Attraction_DP
 - AttractionsByName
 - AttractionsList

Attraction × Attraction_DataProvider * × Attraction_DP ×

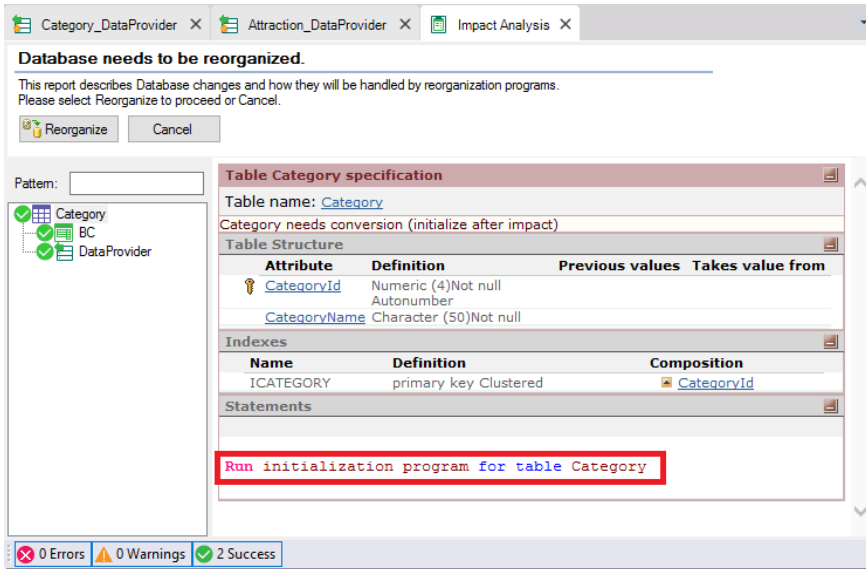
Source * Rules Variables

```

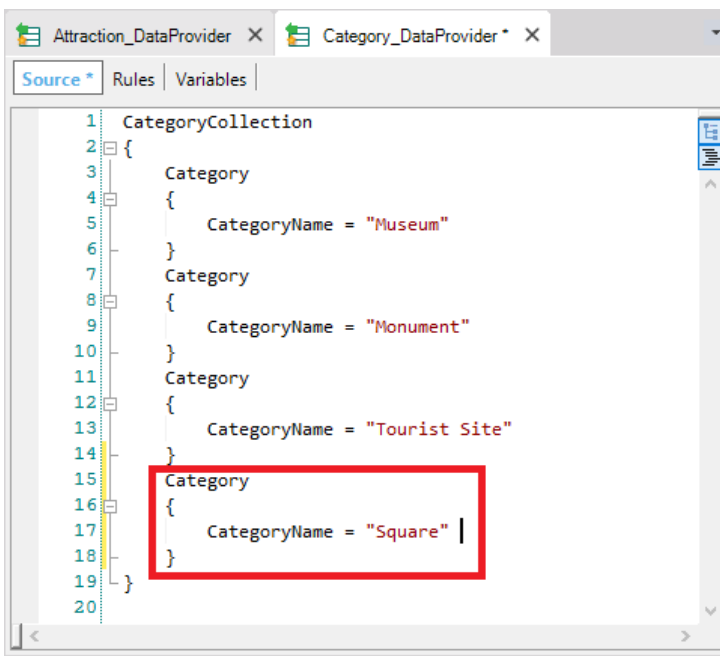
1 AttractionCollection
2 {
3   Attraction
4   {
5     AttractionName = "Louvre Museum"
6     CountryId = find( CountryId, CountryName = "France")
7     CategoryId = find( CategoryId, CategoryName = "Museum")
8     CityId = find( CityId, CityName = "Paris")
9     AttractionPhoto = Attraction_louvre.Link()
10  }
11  Attraction
12  {
13    AttractionName = "The Great Wall"
14    CountryId = find( CountryId, CountryName = "China")

```

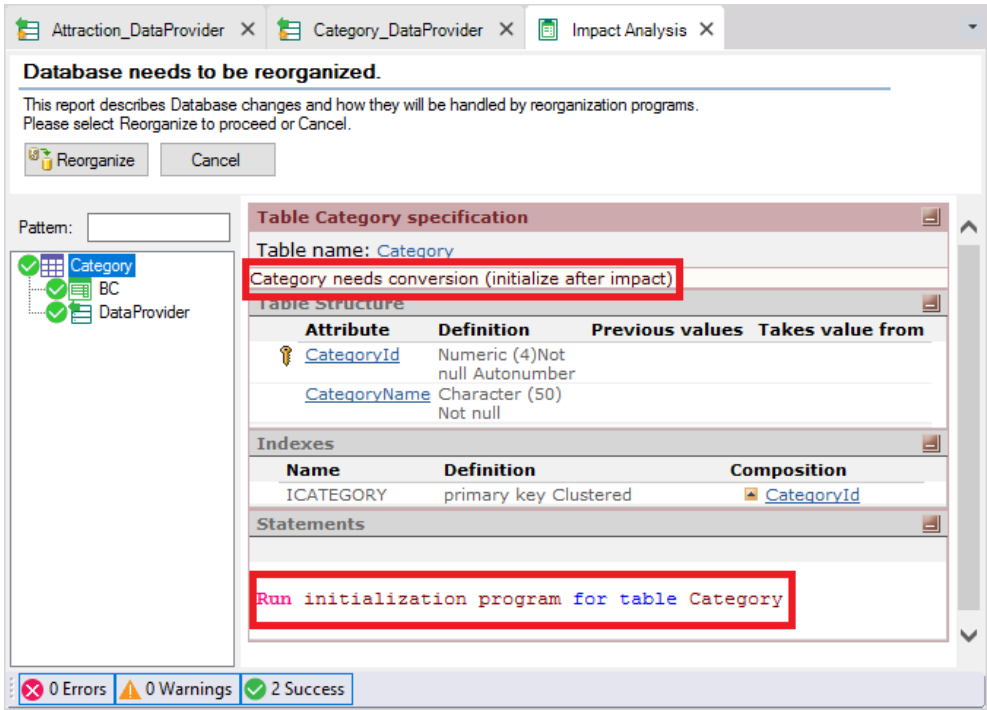
Ejecutemos lo hecho hasta ahora. F5.



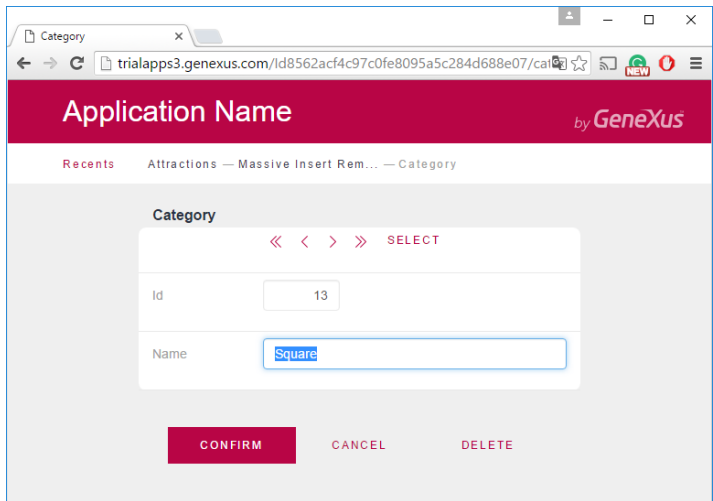
Ahora bien, si ahora o más adelante necesitamos modificar el Data Provider de inicialización, agregando, por ejemplo, una categoría que no habíamos contemplado inicialmente:



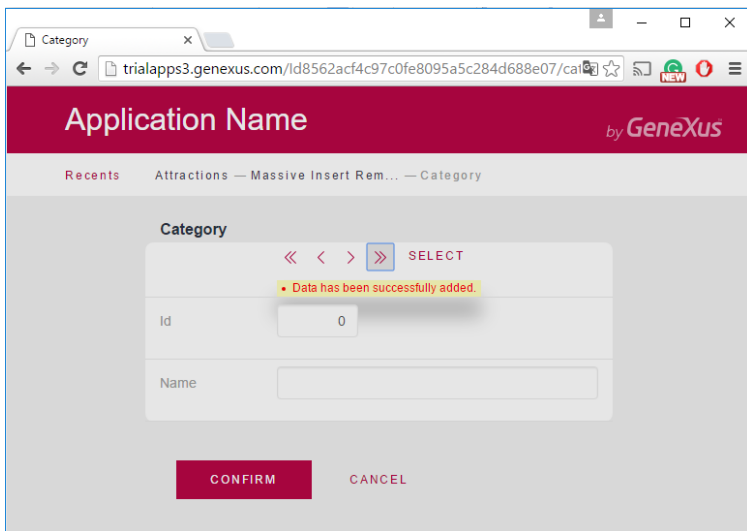
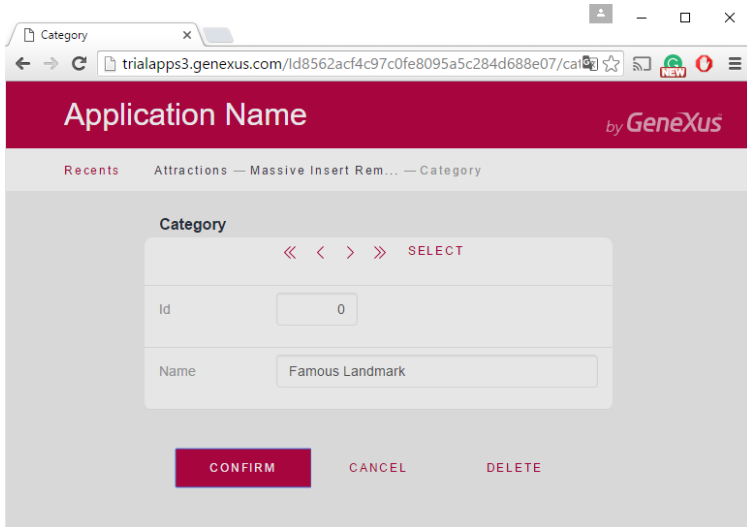
... al hacer F5 GeneXus se dará cuenta de que cambió el data provider y volverá a ejecutarlo.



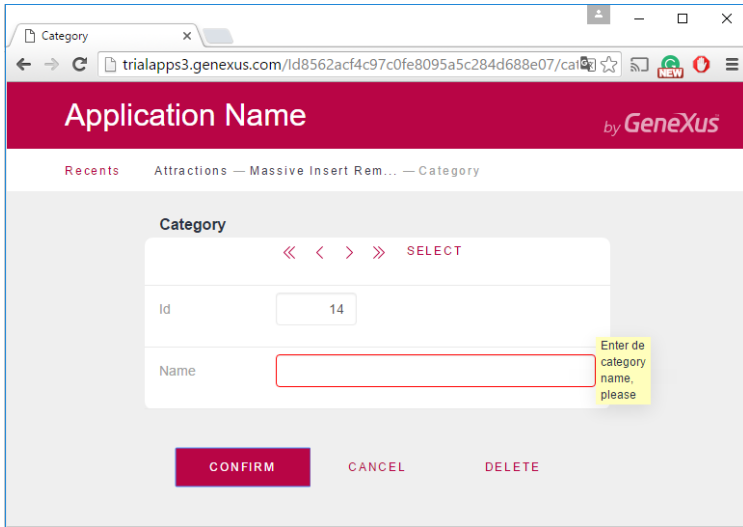
Borremos la tabla antes de reorganizar, para que no nos duplique registros debido al autonumber.



Observemos que la transacción sigue funcionando, en todo lo demás, de manera estándar. Es decir, seguiremos insertando, actualizando y eliminando sus datos a través de la pantalla, como siempre.



... se ejecutarán sus reglas:



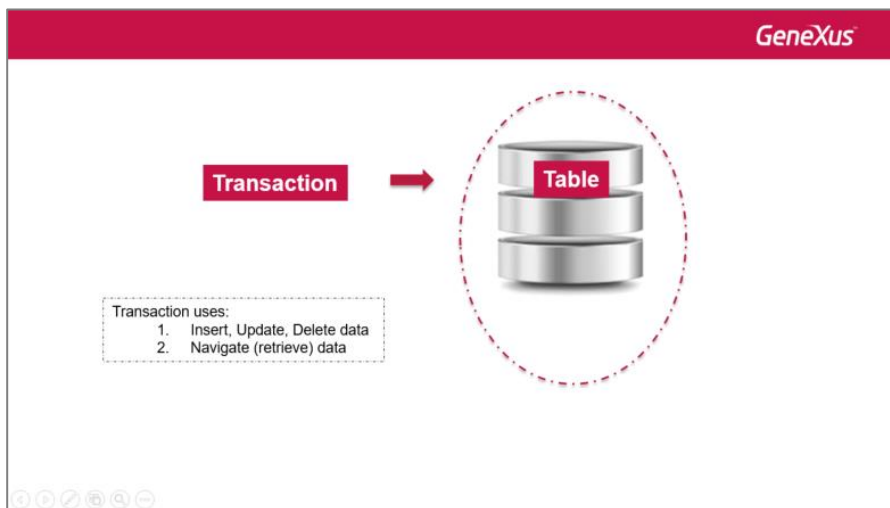
...y se podrá utilizar el business component asociado, como lo hemos hecho antes. Lo que hemos visto solamente afecta a la inicialización de sus datos.

Pero si la transacción corresponde a información que no cambia con el tiempo, como por ejemplo, los países, estados o departamentos de un país, parámetros de un sistema, etcétera, resulta innecesario que la transacción o el business component nos permita actualizar sus datos. Para asegurarnos que los datos no se modifiquen, configuramos la propiedad "Update Policy" en Read Only.

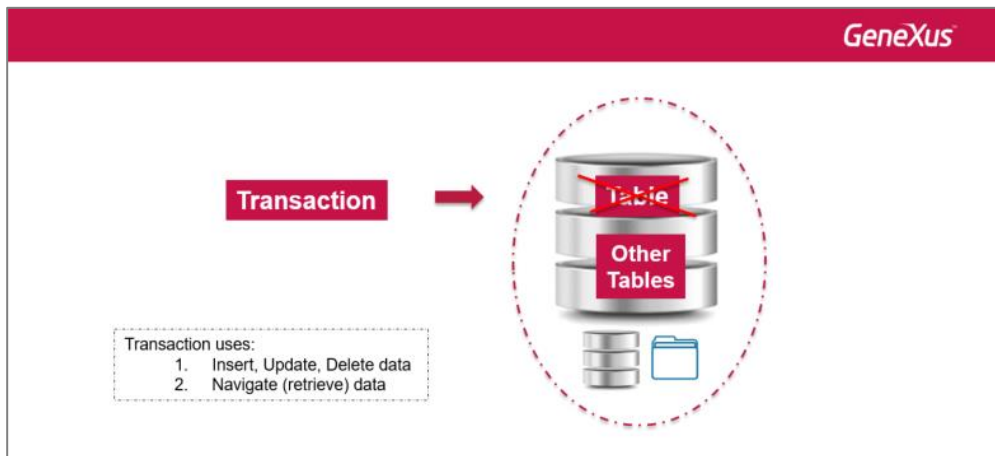
▼ Data

Data Provider	True
Used to	Populate data
Update Policy	Read Only

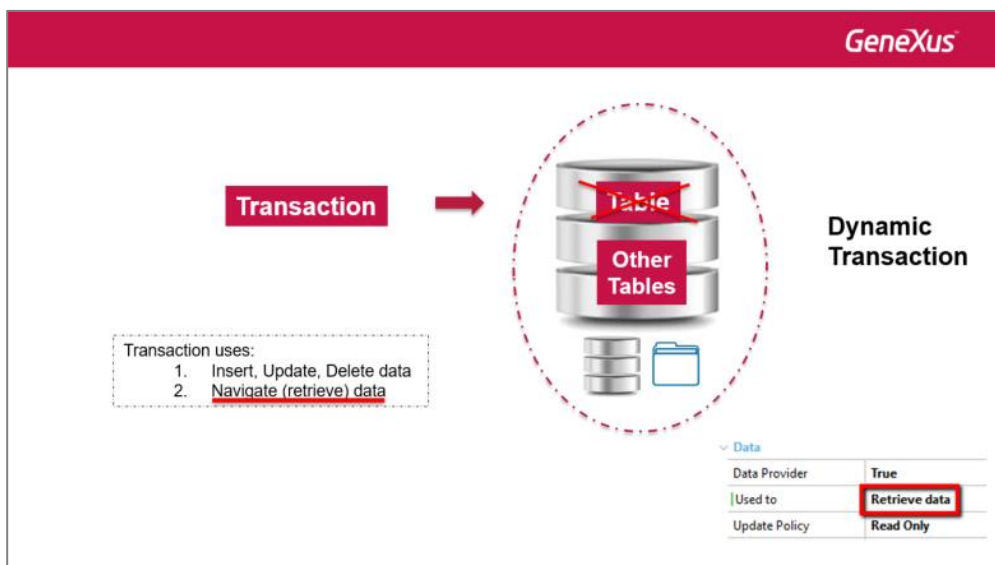
Hasta aquí el uso de la transacción corresponde más o menos al conocido, donde la transacción tiene su "tabla" asociada.



Pero tenemos otros casos en los cuales los datos de la transacción se obtienen de otras fuentes, que pueden ser consultas complejas a la base de datos que incluya ir a buscar a la información a varias tablas, o consultas a otras bases de datos, etcétera.



En este caso la transacción no tendrá esa tabla asociada. El Data Provider es quien se encarga de obtener esos datos. Para especificar este uso, a la propiedad "Used to" le configuramos el valor "Retrieve data".



A estas transacciones se les llama "transacciones dinámicas", sobre las que no ahondaremos en este curso.

En conclusión, para poblar de datos una tabla no lo haríamos en forma manual como hicimos en el video anterior, sino utilizando la propiedad Data Provider de la transacción.

Para finalizar, hagamos un Commit de nuestros cambios en GeneXus Server.

Team Development X

Commit | Update | History | Activity | Versions

Comment

Data Provider used to Populate Data for Category and Attraction transactions

Objects

Pending Commits (5/5) Ignored Objects

<input checked="" type="checkbox"/>		Name	Type	Descri	Modified On	Module	Action	Last Synch	User
<input checked="" type="checkbox"/>		Attraction	Transaction	Attr...	9/7/2016 8:5...	Root Mod...	Modifi...	9/6/2016...	ARTECHIC...
<input checked="" type="checkbox"/>		Attraction_DataP...	Data Provider	Attr...	9/7/2016 8:5...	Root Mod...	Insert...	7/5/2016...	ARTECHIC...
<input checked="" type="checkbox"/>		Category	Transaction	Cate...	9/8/2016 1:5...	Root Mod...	Modifi...	9/6/2016...	ARTECHIC...
<input checked="" type="checkbox"/>		Category_DataPr...	Data Provider	Cate...	9/8/2016 12:...	Root Mod...	Insert...	7/5/2016...	ARTECHIC...
<input checked="" type="checkbox"/>		GeneXus	Module	Gen...	9/8/2016 11:...	Root Mod...	Modifi...	9/5/2016...	GeneXus

Videos training.genexus.com

Documentation wiki.genexus.com

Certifications training.genexus.com/certifications