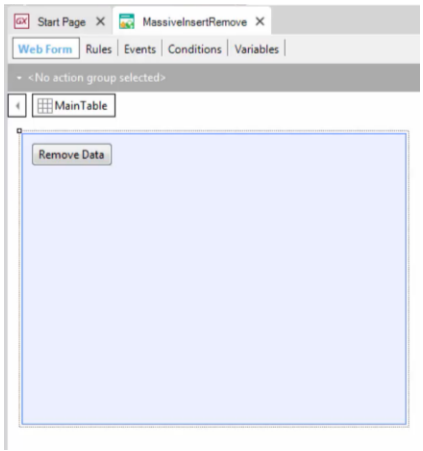
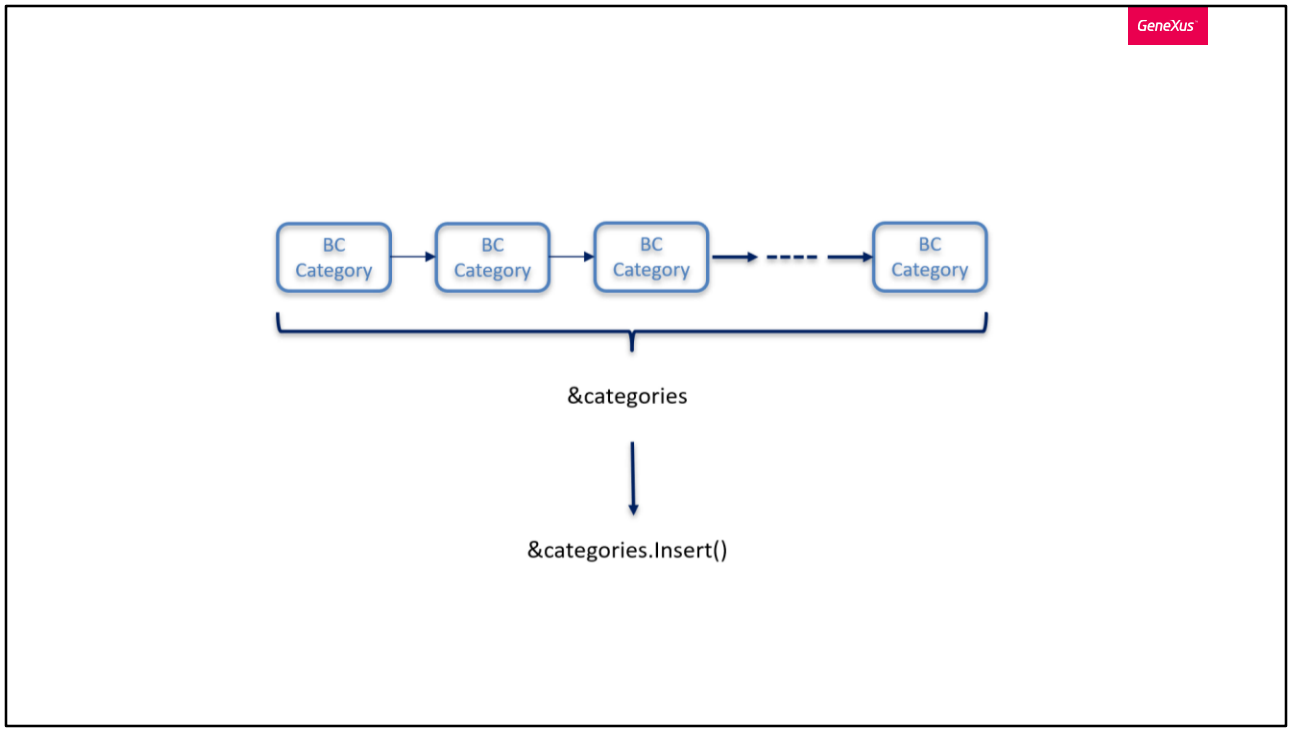
Poblando con datos utilizando Business Component y Data Provider

GeneXus™



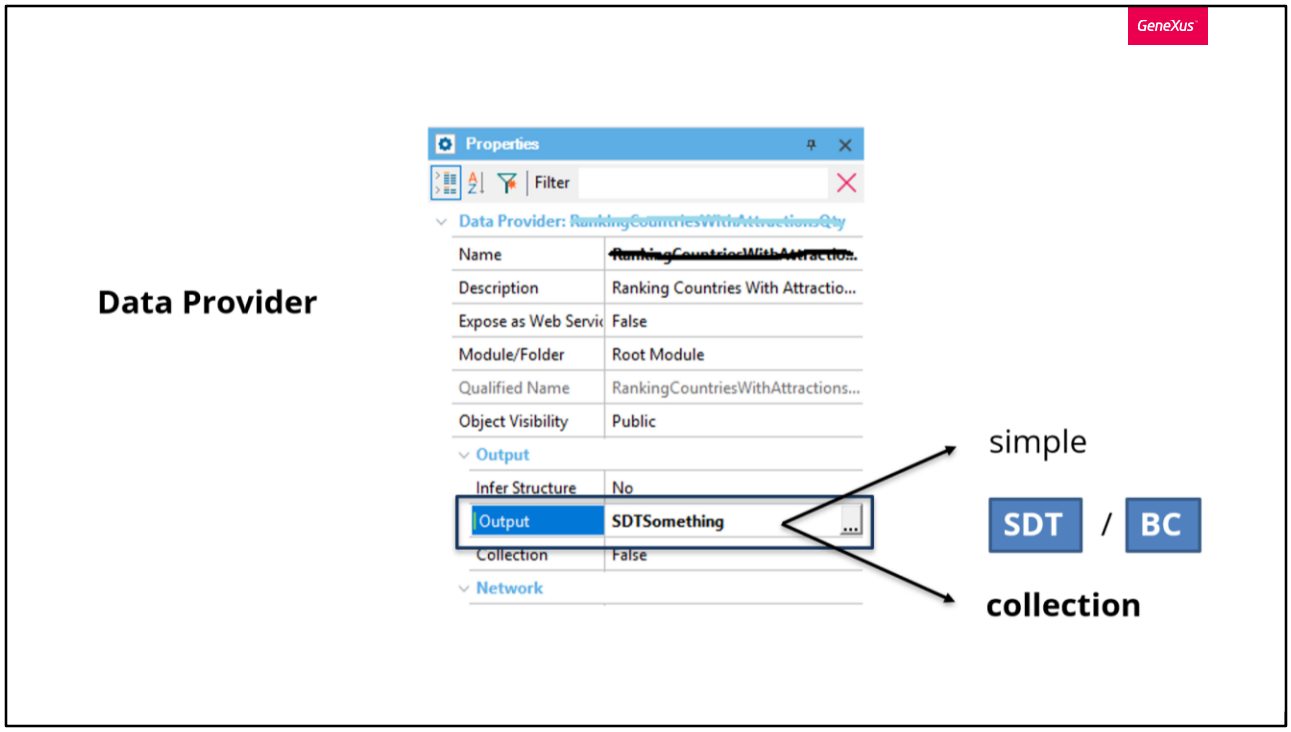
Supongamos que las tablas de países, ciudades, etc. ya tienen datos. Como anteriormente eliminamos los datos de atracciones y categorías, nuestro objetivo será inicializar las tablas de categorías y atracciones, con datos, para partir de tablas no vacías.

Para ello abrimos el web panel “MassiveInsertRemove”, creado en el video anterior, le agregamos un botón “Initialize data” y haremos uso del data provider en conjunción con los business components recién estudiados.



Si lográramos obtener una variable colección de ítems del tipo business component de Category, cargada con las categorías que deseamos ingresar a la base de datos, nos alcanzará con aplicar el método Insert() a esa variable colección, pues como mencionamos en el video anterior, esto permitirá hacer el Insert de todos los ítems, es decir, de todos los business components de la colección.

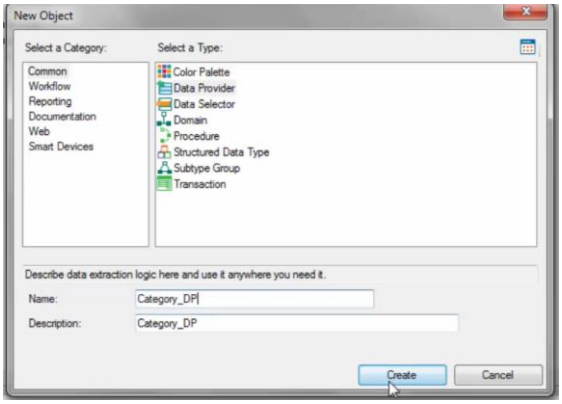
Nuestro problema se reduce, entonces, a obtener esa colección. ¿Cómo hacemos?



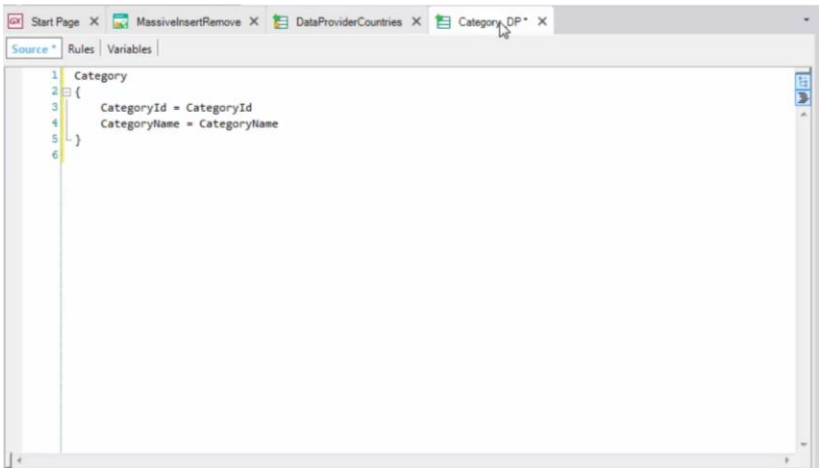
Hasta ahora sabíamos que un Data Provider nos permitía devolver datos estructurados, tanto simples como colección.

En nuestro caso, querremos devolver una colección de categorías, pero estas categorías no son tipos de datos estructurados sino **business components**.

Sin embargo, un business component es exactamente igual que un SDT en lo que hace a su estructura. Por lo que los data providers también nos permitirán cargar y devolver business components, tanto simples como colecciones.

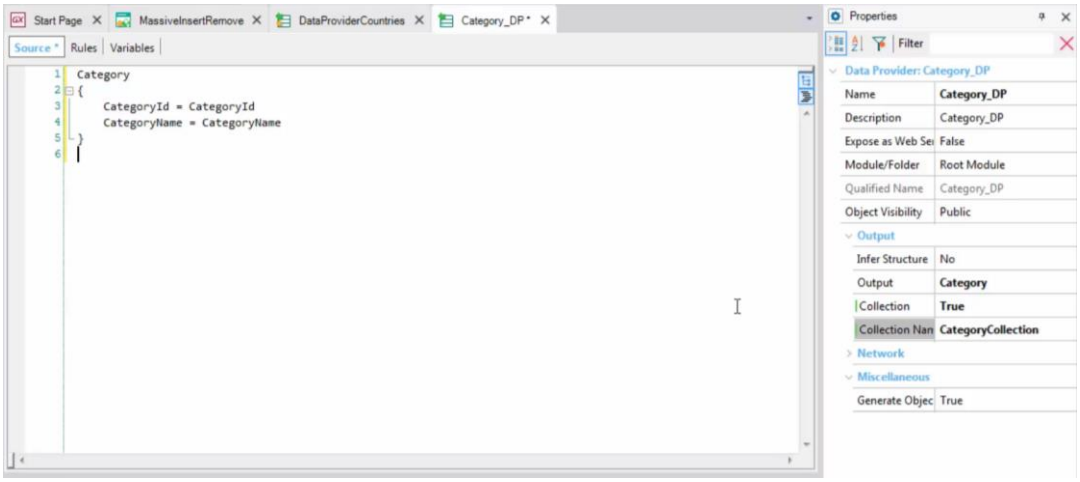


Por allí vendrá nuestra solución. Creemos un data provider para cargar las categorías. Lo llamaremos Category_DP.



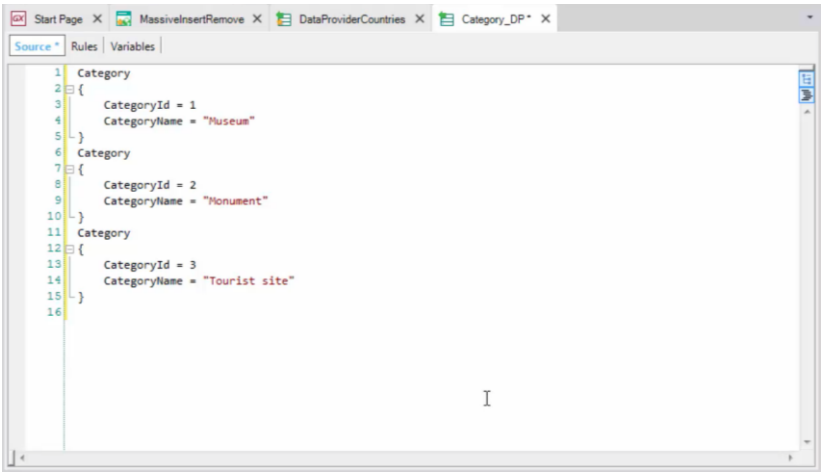
Arrastramos la transacción Category dentro del Source del data provider y vemos que nos escribe la estructura de la transacción. Notemos que a la izquierda tenemos los elementos del business component, que estarán en memoria, y que tienen el mismo nombre que los atributos, pero no lo son.

Mientras que a la derecha, por defecto se están mostrando, esta vez sí, los atributos de la tabla correspondiente, desde donde el data provider obtendrá los datos para cargar el BC que está en memoria.
Si esto es lo que quisiéramos, el data provider debería devolver una colección de este business component, dado que la tabla tiene muchos registros.



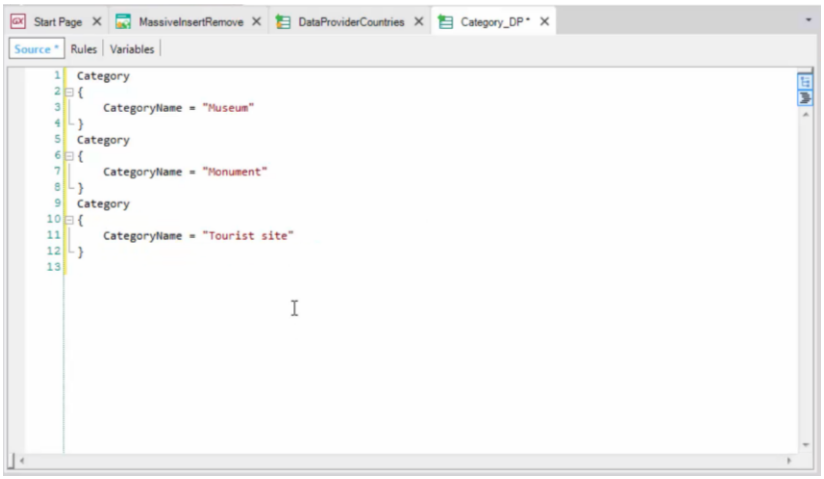
Si vamos a las propiedades, vemos que la propiedad Output asumió el valor del business component, pero la propiedad collection no está en True, como necesitamos.

Así que la cambiamos y nos aparece la nueva propiedad Collection name, que por defecto asume el nombre del data provider. Lo cambiamos por CategoryCollection.

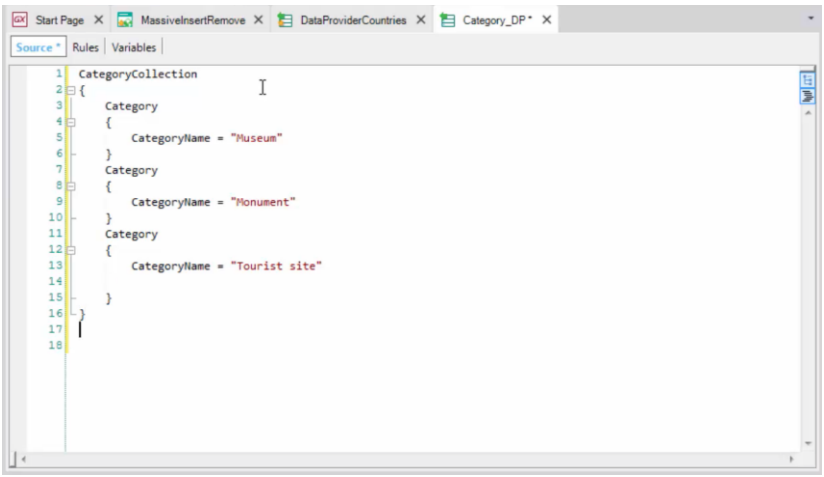


Y además, nosotros no queremos cargar esta colección con datos de la base de datos, sino que queremos asignarles valores nuevos, especificados por nosotros.

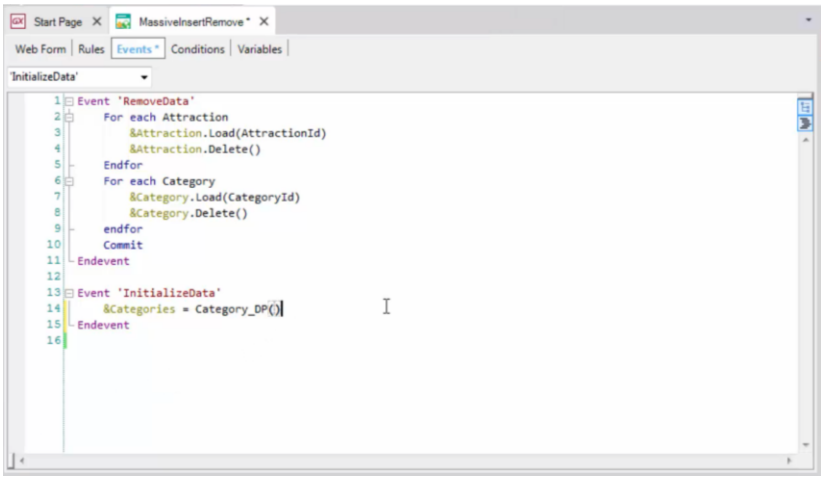
Por lo tanto, uno a uno, escribiremos grupos asociados a los ítems de la colección:



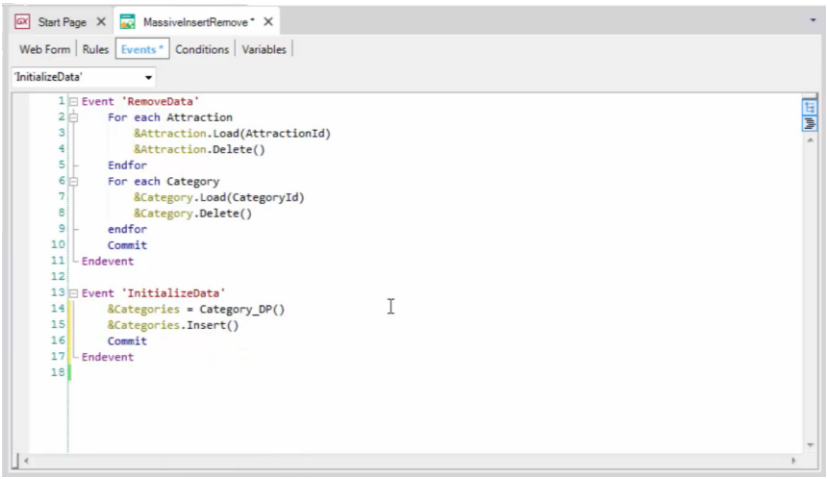
Como CategoryId es un atributo autonumerado, no necesitamos asignarle valor cuando queramos insertar un registro, que será lo que haremos luego, así que directamente borramos esa asignación:



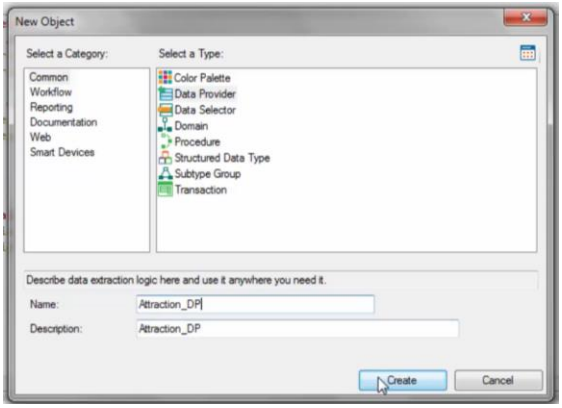
Y como lo que queremos devolver es una colección de nombre CategoryCollection, aunque no sea necesario --porque al ponerle a la propiedad Collection el valor True, el data provider ya sabe que devolverá una colección, a la que llamará de esa manera--, para clarificar el código podemos explicitar lo que GeneXus ya interpreta: encerrando todos los grupos Category dentro del grupo CategoryCollection, que corresponde a la colección.



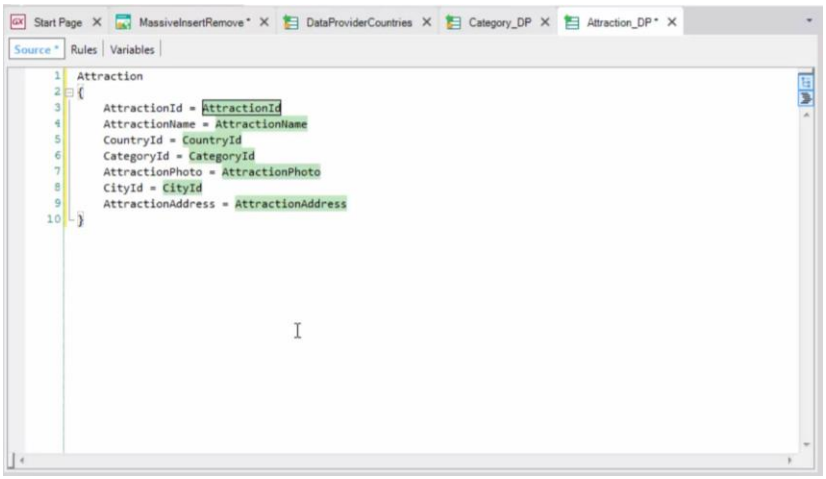
Ahora solamente nos faltará invocar a este Data Provider desde el evento asociado al botón del webpanel:



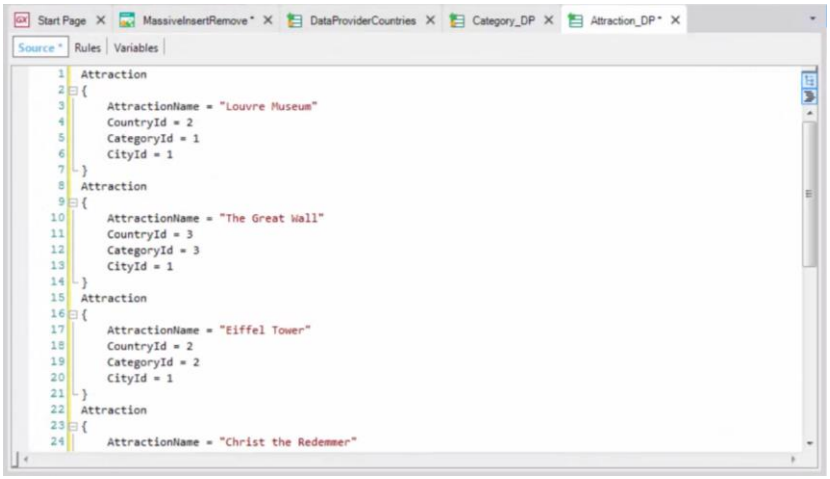
y luego insertar en la base de datos:



Ahora tendríamos que inicializar la tabla de atracciones. Análogamente crearemos un data provider, Attraction_DP.

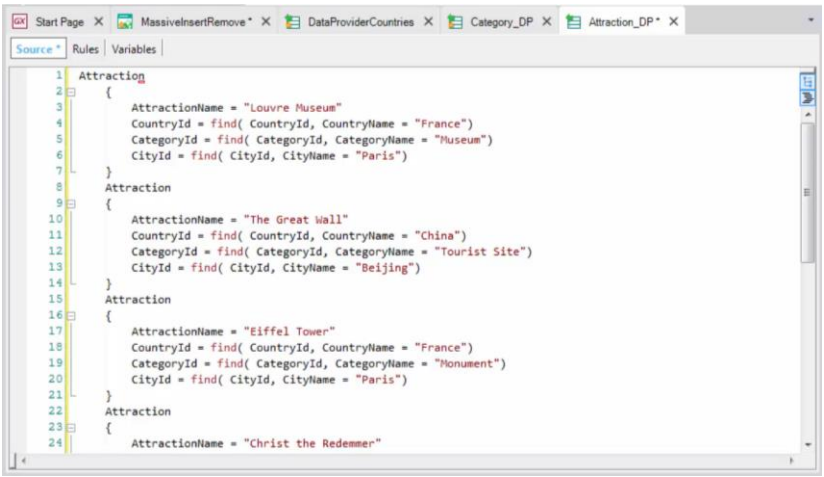


Arrastraremos la Transacción (de la que ya habíamos obtenido el business component) y vemos que cada elemento del business component queda inicializado por defecto con el atributo correspondiente de la tabla.



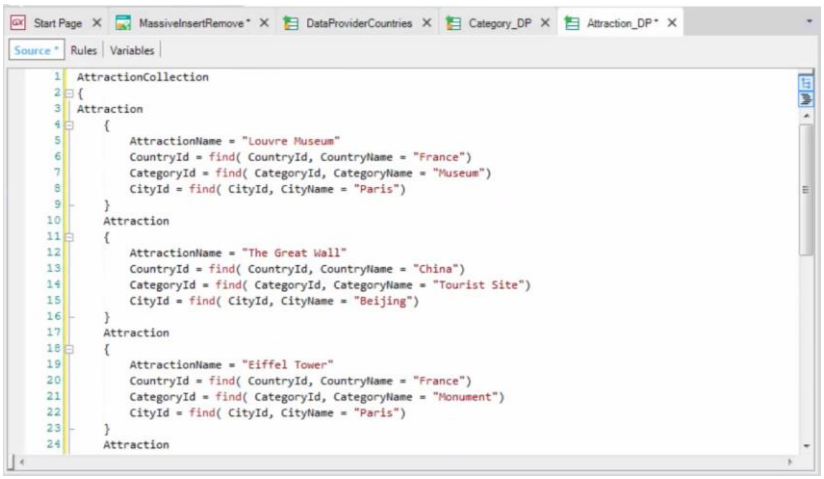
Otra vez, vemos que solamente se toman en cuenta los atributos presentes físicamente en la tabla, no los que son inferidos en la transacción o fórmulas.

Como no nos interesa cargar atracciones existentes (pues de hecho ejecutamos este data provider para cargar los primeros datos), eliminamos todos estos atributos, e ingresaremos a mano estos valores. Además, como el Id es autonumber, tampoco necesitamos asignar valor para este elemento del business component. Las fotos de las atracciones las asignaremos después, así que también quitamos este atributo.



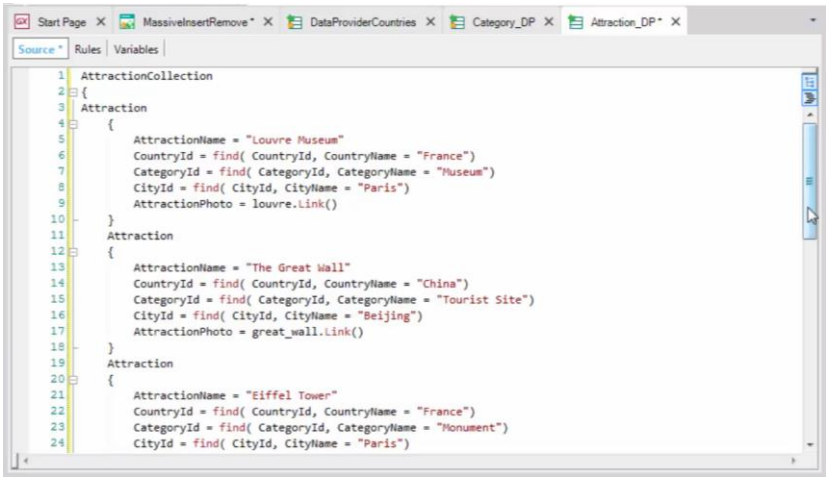
Como aquí estamos asignando los valores de CountryId, CityId y CategoryId de memoria, quizás no existan en las tablas correspondientes. Si alguno de los valores no existiera, al momento de intentar insertar los registros con el business component se dispararán los controles de integridad referencial correspondientes y la inserción fallará.

Para evitar asignar valores que pueden no existir, vamos a utilizar la fórmula Find para encontrar los identificadores correctos a partir de los nombres del país, ciudad o categoría.



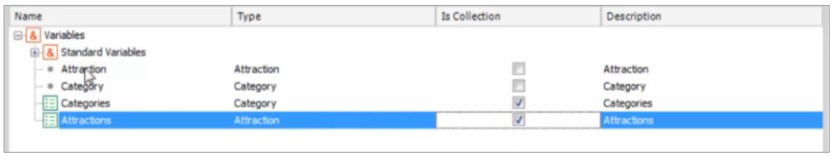
Notemos que las fórmulas Find están accediendo a la base de datos solamente a buscar los identificadores correspondientes a los nombres que usamos, pero el resto de los valores asignados al business component son fijos.

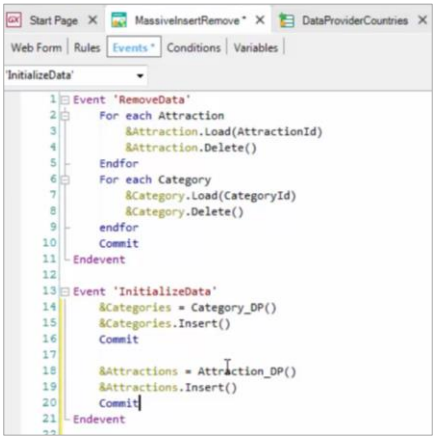
Al igual que hicimos con el Data Provider de las categorías, debemos poner la propiedad Collection en True, ya que vamos a devolver muchas atracciones y también ajustaremos la notación en el source encerrando los grupos dentro del grupo AttractionCollection, para indicar que es una colección de atracciones.



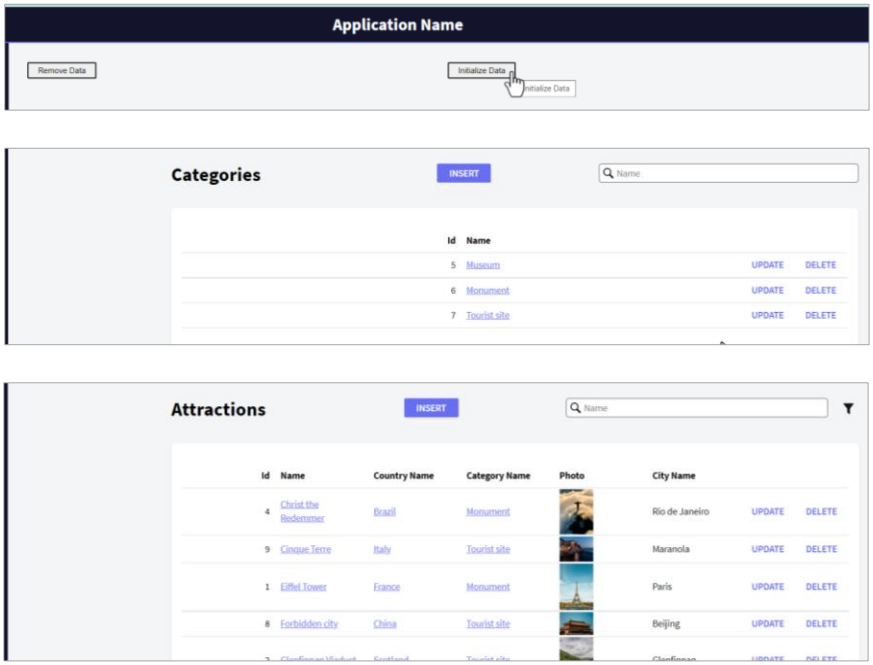
Para cargar también las fotos de las atracciones, una posibilidad es insertarlas primero como objetos imagen en la KB…

Y luego para cada grupo del Data Provider, simplemente asignarle a AttractionPhoto el nombre de la imagen punto link.

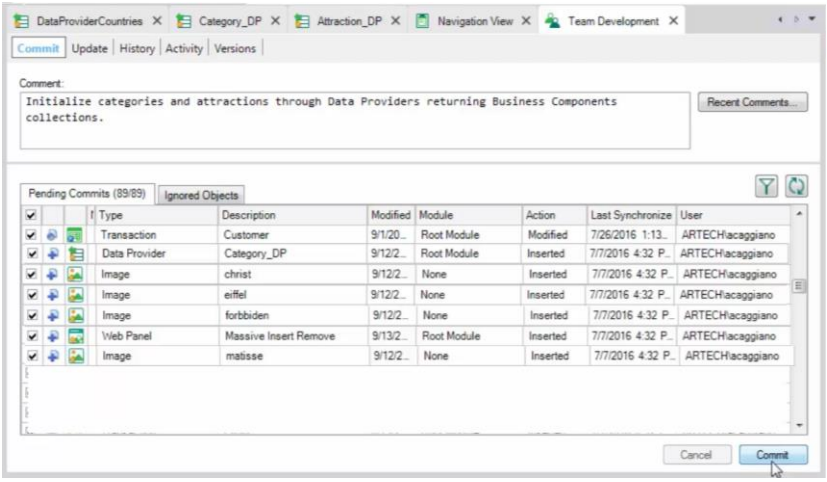




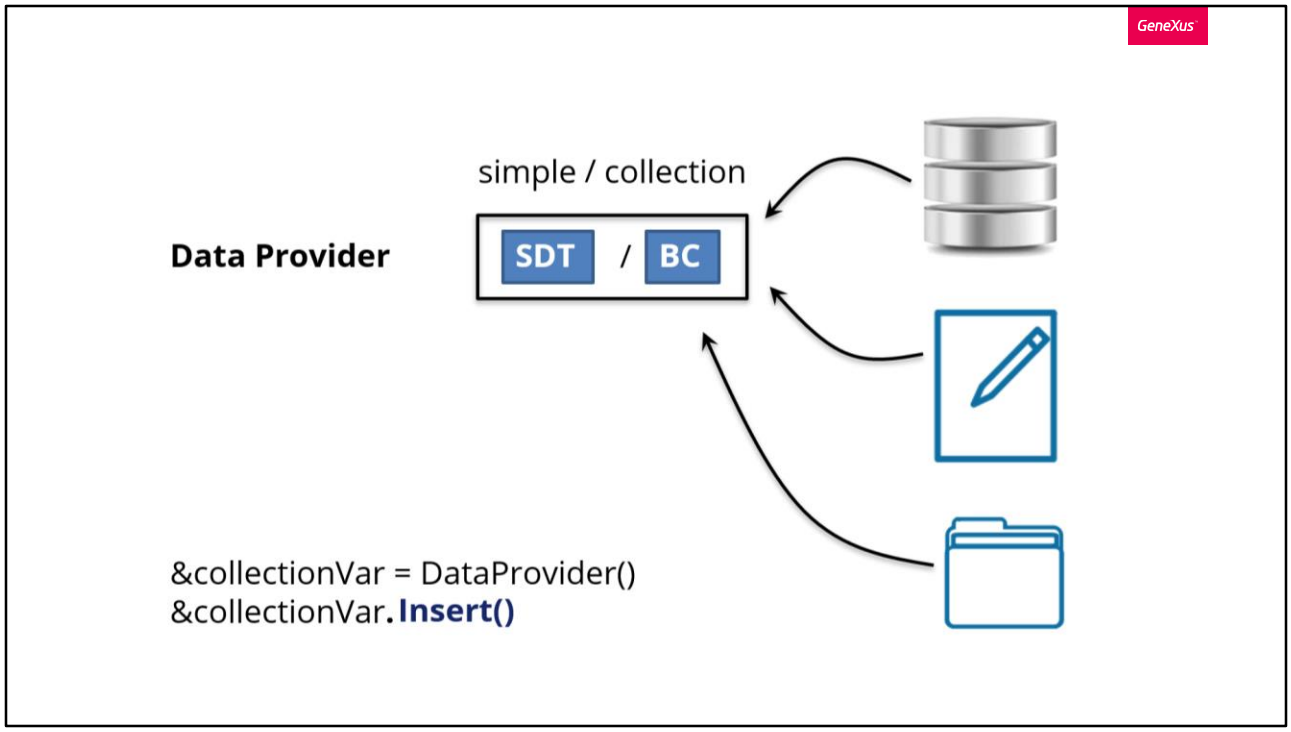
Ahora solamente falta invocar al Data Provider para que devuelva la colección cargada….



Observemos que para poder insertar las atracciones, primero tenemos que tener creadas las categorías, por lo que el orden es el que usamos en el código del evento.



Hagamos un commit en GeneXus Server.



En este video vimos cómo un data provider no solamente permite cargar una estructura con datos de la base de datos… sino también a partir de datos fijos… pero también podrá hacerlo a través de otras fuentes externas, como podrá estudiar en cursos más avanzados.

Además vimos que un data provider permite cargar la estructura de un business component (y no sólo de un SDT), tanto simple como colección.

Por último vimos que en caso de que la estructura sea del tipo colección podremos aplicar métodos que afectan a todos los items de la colección, en una sola operación, como por ejemplo insert() y delete().

GeneXus™

training.genexus.com
wiki.genexus.com