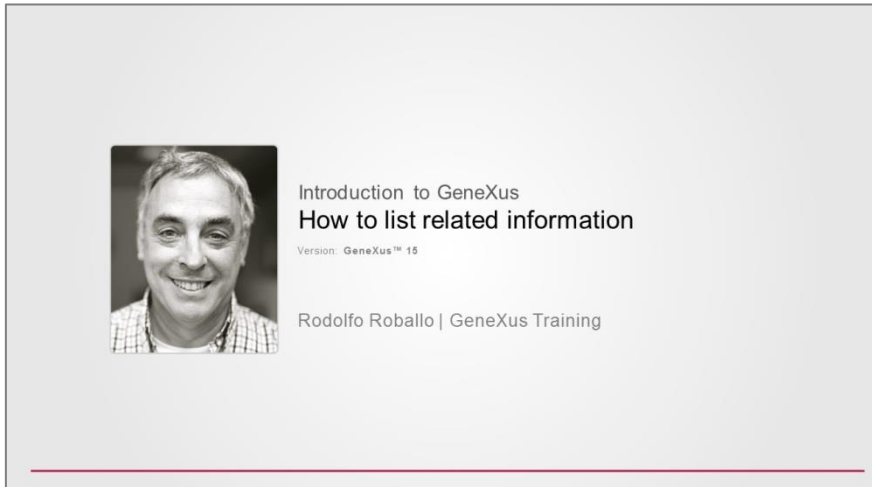



## Cómo listar información relacionada





En el video anterior habíamos visto cómo definir un procedimiento que contenía en su Source un único comando For each, que accedía a los datos de una tabla base (y su tabla extendida).

Ahora supongamos que la agencia de viajes nos solicita un listado que muestre todas las categorías de atracciones turísticas y para cada categoría, todas sus atracciones.





### Categories and their Attractions


Category: Museum

Id	Name	Country	Photo
1	Louvre Museum	France	
5	Smithsonian Institute	United States	

Category: Monument

Id	Name	Country	Photo
3	Eiffel Tower	France	
4	Christ the Redemmer	Brazil	

Category: Famous Landmark

Id	Name	Country	Photo
2	The Great Wall	China	

Antes de resolver este requerimiento vamos a ejecutar la aplicación para agregar una categoría más a las que teníamos y un par de atracciones turísticas. Presionamos F5...

Agregamos esta categoría:

Category	
	SELECT
Id	3
Name	Famous Landmark
CONFIRM CANCEL	






Y luego a la Muralla china le asociamos la categoría recién creada...

Attraction	
Id	2
Name	The Great Wall
Country Id	3
Country Name	China
Category Id	3
Category Name	Famous Landmark
Photo	
City Id	
City Name	Beijing

Y agregamos al Cristo Redentor...

Al país United States agreguémosle la ciudad "Washington":

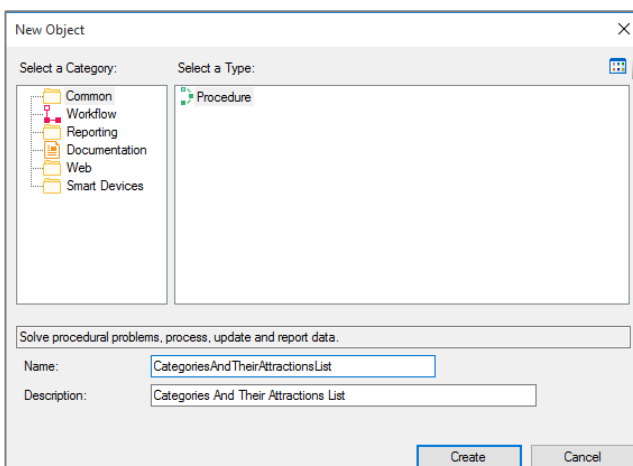
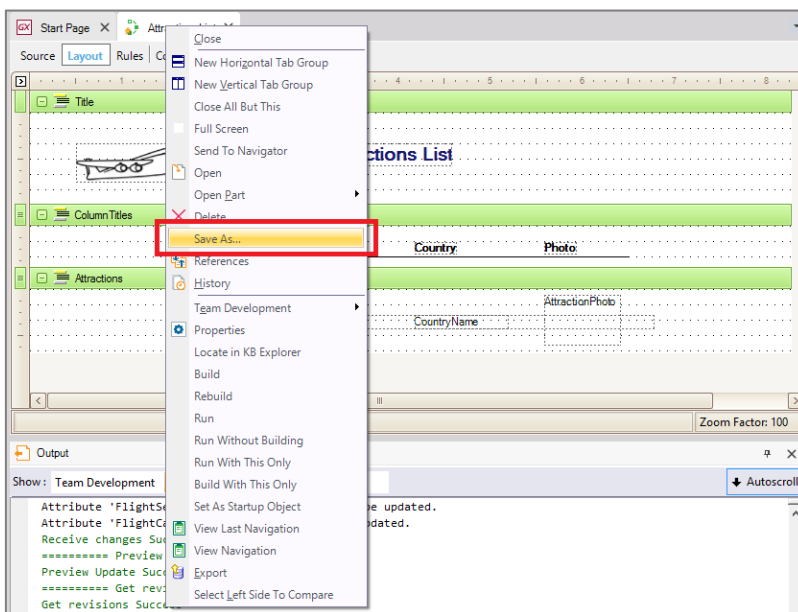
Para agregar al Instituto Smithsonian...

Id	Name	Country Name	Category Name	Photo	City Name		
4	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE
3	Eiffel Tower	France	Monument		Paris	UPDATE	DELETE
1	Louvre Museum	France	Museum		Paris	UPDATE	DELETE
5	Smithsonian Institute	United States	Museum		Washington	UPDATE	DELETE
2	The Great Wall	China	Famous Landmark		Beijing	UPDATE	DELETE

Ahora sí, pasemos a definir el listado que nos solicitan.

Creamos un nuevo objeto de tipo procedimiento.

Para no hacer todo de cero, salvemos el que teníamos con otro nombre.



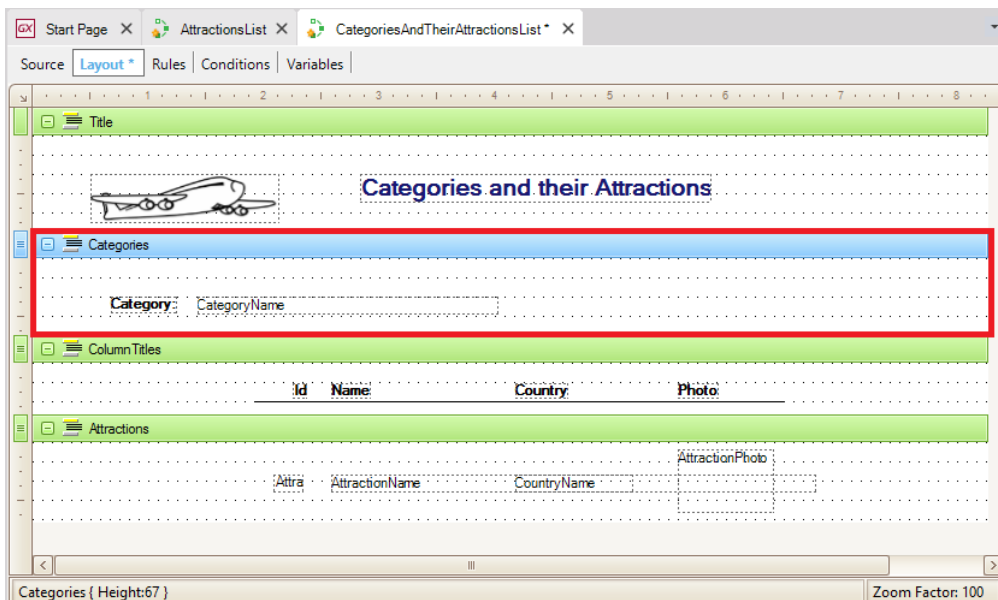
Borrémosle el contenido del Source y modifiquemos el Layout...

Cambiamos el texto de este Text Block



Y agreguemos un print block para mostrar las categorías. Si bien podemos ubicarlo en cualquier lugar, porque el orden de impresión de los printblocks lo daremos en el Source, vamos a colocarlo donde queremos que salga listado, así nos simplifica ver cómo se mostrará todo en ejecución.

Llamaremos Categories a este nuevo printblock. E insertaremos allí un Text Block... Category, ...y un atributo... CategoryName. Los demás print blocks quedarán como están.



Observemos que tenemos dos print blocks con contenidos fijos: Title y ColumnTitles y dos print blocks con contenidos variables, que habrá que extraer de la base de datos: Categories y Attractions. Ambos contienen atributos. Categories tiene a CategoryName, de la tabla CATEGORY, y Attractions contiene a todos estos atributos, que ya habíamos visto que pertenecían a la tabla extendida de ATTRACTION.

Ahora pasemos al Source.

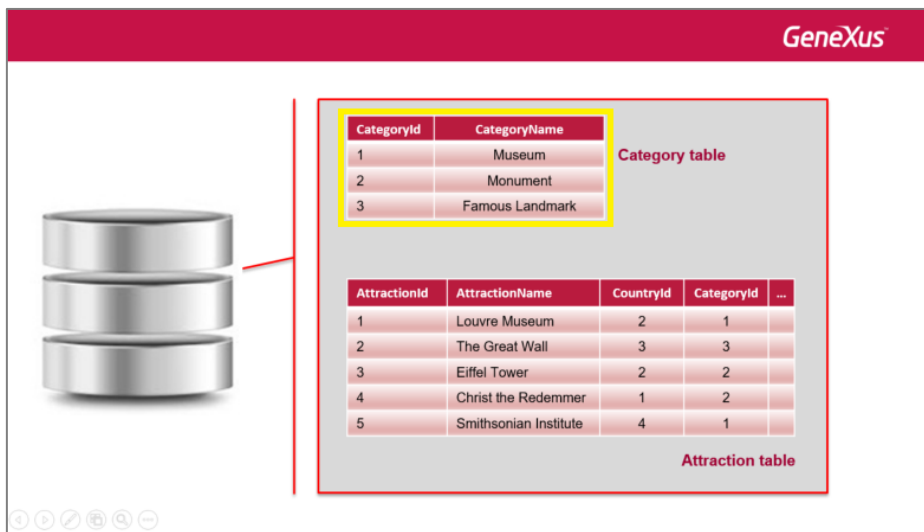
Escribamos la primera instrucción para imprimir el título del listado, igual que como habíamos hecho antes.

```
Source * | Layout * | Rules | Conditions | Variables |
Subroutines
1 print Title
2 |
3 |
4 |
```

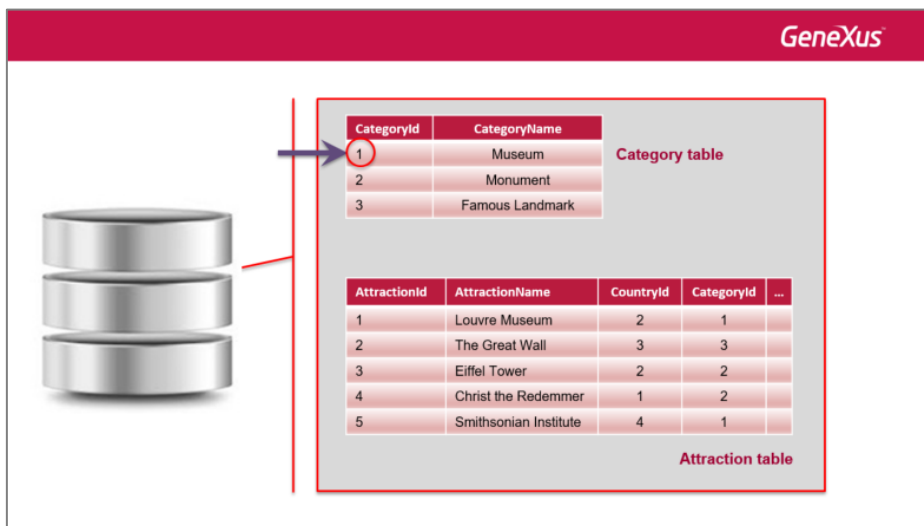
Este print block no contiene atributos, por lo que no hay que ir a buscar información a la base de datos y es por eso que lo podemos mandar a imprimir sin más.

Y ahora detengámonos a ver cómo seguimos...

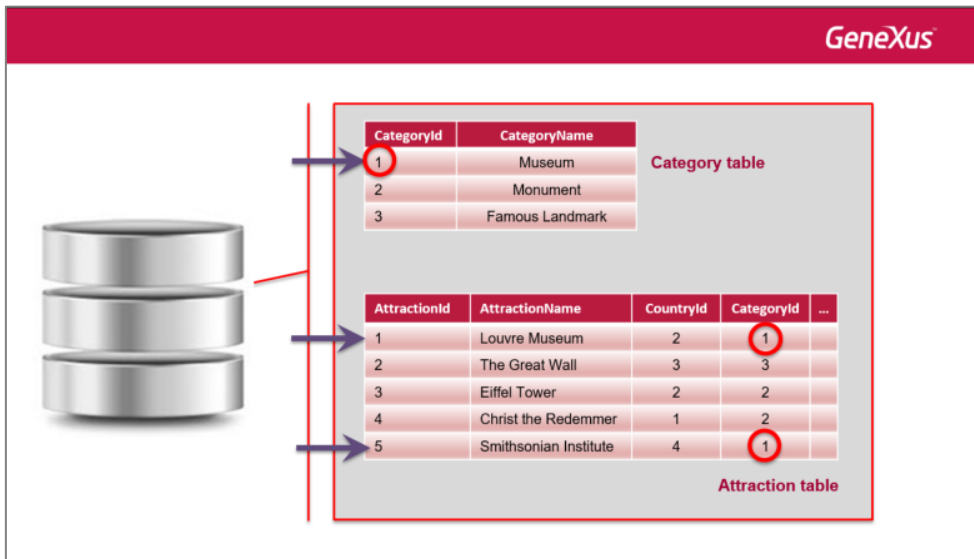
Como tenemos que navegar las categorías



y para cada una de ellas...

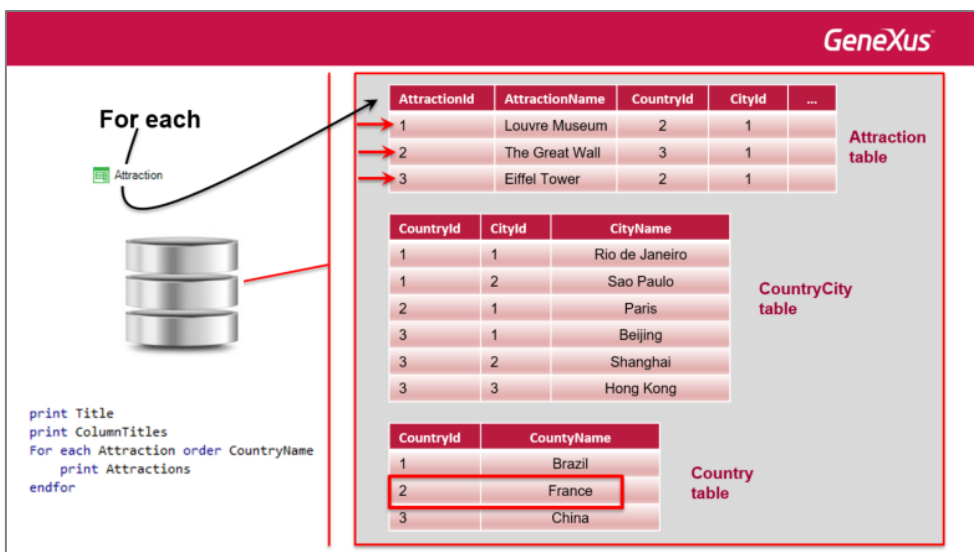


navegar varias atracciones (las que pertenecen a dicha categoría)

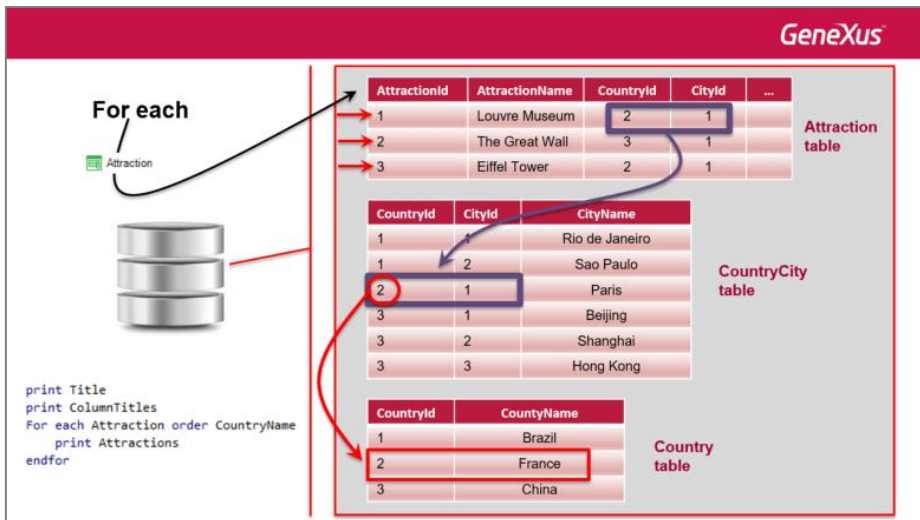


este listado es diferente del que hemos resuelto anteriormente.

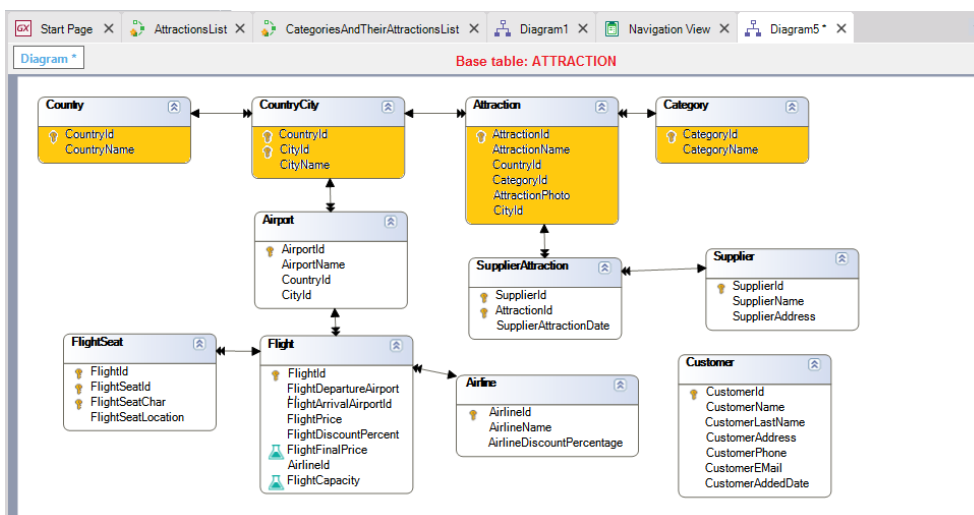
En el listado anterior navegábamos las atracciones



y dado que cada atracción tiene un sólo país, podíamos recuperar el nombre del país de cada atracción, por estar en la tabla extendida de la tabla base que estábamos navegando.

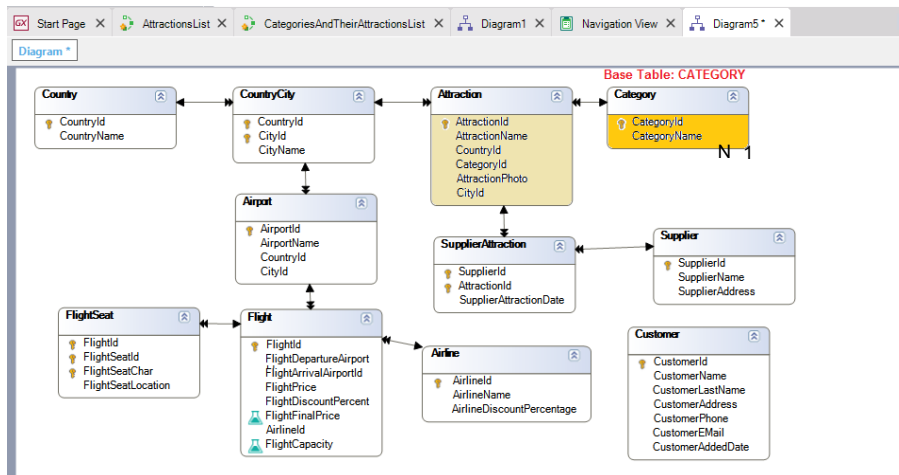


Siempre que la información que queremos recuperar esté disponible en la tabla extendida de la tabla base que estamos navegando, entonces



la podemos referenciar directamente en el mismo For each. Era el caso de CountryName

En cambio, si estamos navegando una tabla –en nuestro caso Category– y necesitamos para cada registro accedido



navegar **varios registros relacionados** que están en otra tabla que no pertenece a la tabla extendida de la tabla que estamos navegando o recorriendo, como es nuestro caso ahora, con la tabla ATTRACTION, entonces necesitaremos escribir **otro For each dentro del primero, para recorrer el conjunto de registros relacionados.**

Es decir, tendremos un for each **anidado** dentro del otro.

Volvamos entonces al Source de nuestro procedimiento, y comencemos a escribir un primer For each para navegar y mostrar las categorías

```

1 Print Title
2 For Each
3

```

Al lado del For each escribimos Category... Recordemos que aquí va la **transacción base**, es decir, el nombre del nivel de la transacción cuya información queremos navegar.

¿Qué es lo primero que queremos hacer con cada categoría accedida por el for each? Imprimirla. Dentro del For each entonces, escribimos: Print Categories

Y como el printblock Categories solamente tiene incluido el atributo CategoryName, y GeneXus dedujo que la **tabla base** del for each es CATEGORY y CategoryName está incluido en la tabla extendida de esa tabla base (porque en este ejemplo está en la propia tabla), entonces estará todo en orden y se podrá recuperar la información correctamente. En caso contrario GeneXus dará un error.

Luego de imprimir la categoría, queremos navegar el conjunto de **sus** atracciones turísticas ... por lo tanto necesitamos escribir el segundo For each, para recorrer las N atracciones pertenecientes



a la categoría que estamos navegando.

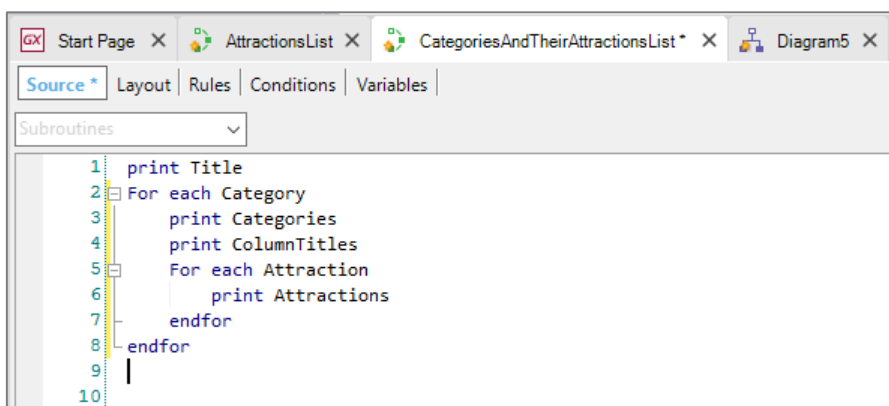
Pero justo antes de navegar el conjunto de atracciones de la categoría, tendremos que mostrar los títulos que encabezarán a estas atracciones que vamos a mostrar, así que escribamos la instrucción **Print AttractionsColumTitles**

Y ahora sí escribamos el 2do For each, dentro del cuerpo del 1ero...

A continuación escribimos Attraction, por ser el nombre de la transacción cuya tabla asociada queremos navegar ahora, y dentro del For each, Print Attractions

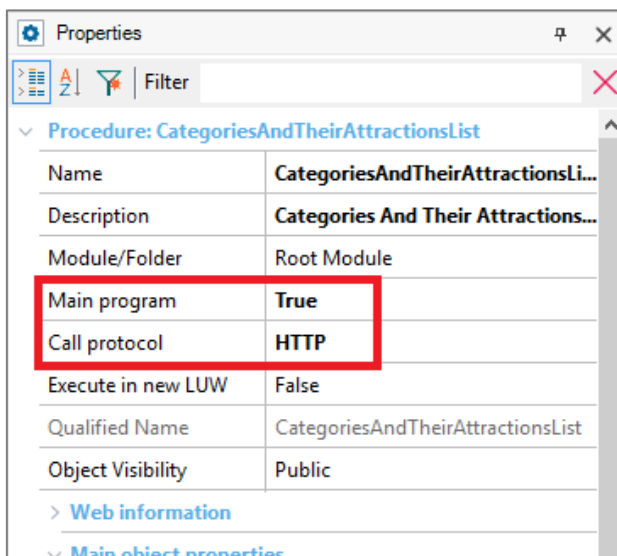
Luego, **Endfor** para cerrar esta navegación

y otra vez **Endfor** para cerrar la primera.



```
1 print Title
2 For each Category
3     print Categories
4     print ColumnTitles
5     For each Attraction
6         print Attractions
7     endfor
8 endfor
9
10
```

Recordemos que para que salga el listado con formato PDF, tenemos que configurar la propiedades del reporte "Main program" con el valor True... "Call protocol" como "HTTP"...



y tenemos que insertar la regla OutputFile ...

Como habíamos grabado el listado anterior con otro nombre, ya estaban todas estas cosas configuradas. En esta regla cambiemos el nombre del pdf por el nuevo:

A screenshot of a software interface with a code editor. The window title bar shows 'GX Start Page', 'AttractionsList', 'CategoriesAndTheirAttractionsList\*', and 'Diagram5'. The 'Rules' tab is active, showing a code editor with the following code:



```
1 Output_file('CategoriesAndTheirAttractions.pdf', 'pdf');  
2  
3
```

Salvamos...



Y vamos a ejecutar este listado... F5

A screenshot of a web application titled 'Categories and their Attractions'. It displays three categories of attractions, each with a table of data and a representative photo.


**Category: Museum**

Id	Name	Country	Photo
1	Louvre Museum	France	
5	Smithsonian Institute	United States	


**Category: Monument**

Id	Name	Country	Photo
3	Eiffel Tower	France	
4	Christ the Redemmer	Brazil	

**Category: Famous Landmark**



Id	Name	Country	Photo
2	The Great Wall	China	

Observamos que se listaron todas las categorías que habíamos ingresado





### Categories and their Attractions


Category: **Museum** ←

Id	Name	Country	Photo
1	Louvre Museum	France	
5	Smithsonian Institute	United States	


Category: **Monument** ←

Id	Name	Country	Photo
3	Eiffel Tower	France	
4	Christ the Redemmer	Brazil	

Category: **Famous Landmark** ←



Id	Name	Country	Photo
2	The Great Wall	China	

y para cada una de ellas se listaron las atracciones que pertenecen a esa categoría.





### Categories and their Attractions


Category: Museum

Id	Name	Country	Photo
1	Louvre Museum	France	
5	Smithsonian Institute	United States	

Category: Monument

Id	Name	Country	Photo
3	Eiffel Tower	France	
4	Christ the Redemmer	Brazil	

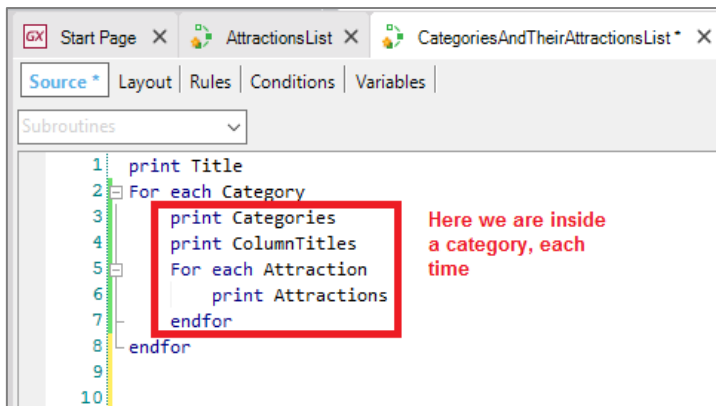
Category: Famous Landmark

Id	Name	Country	Photo
2	The Great Wall	China	

Con esto hemos resuelto el requerimiento pedido por la agencia de viajes.

Ahora bien, **¿cómo supo GeneXus cuáles atracciones debía mostrar para cada categoría si nosotros no le indicamos explícitamente nada al respecto?**

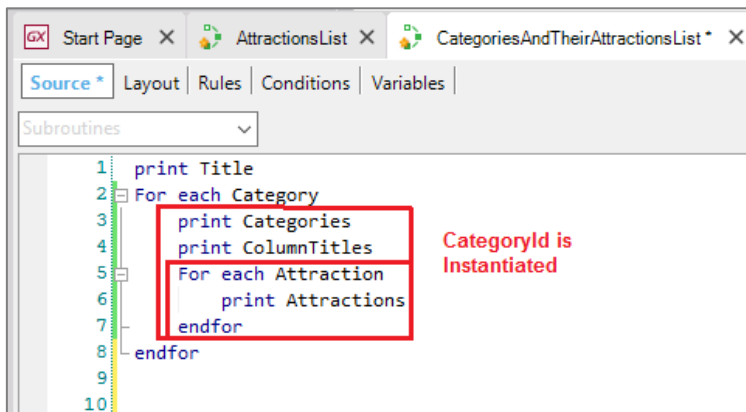
Observemos los for eachs. Sabemos que un for each recorre N registros de una tabla y por cada uno de ellos, ejecuta una serie de instrucciones; las que se encuentran dentro del for each. Dentro de ese “cuerpo” del primer for each, cada vez estaremos posicionados en una y solo una categoría. Decimos que la categoría está **instanciada**, cada vez. Es la categoría **tal**. Cuando se terminan de ejecutar las instrucciones del cuerpo, recién allí se pasa a la siguiente categoría.



```
1 print Title
2 For each Category
3   print Categories
4   print ColumnTitles
5   For each Attraction
6     print Attractions
7   endfor
8 endfor
9
10
```

Here we are inside a category, each time

Por lo que antes de empezar a ejecutar el For each anidado, GeneXus ya sabe en qué categoría está posicionado en ese momento.



```
1 print Title
2 For each Category
3   print Categories
4   print ColumnTitles
5   For each Attraction
6     print Attractions
7   endfor
8 endfor
9
10
```

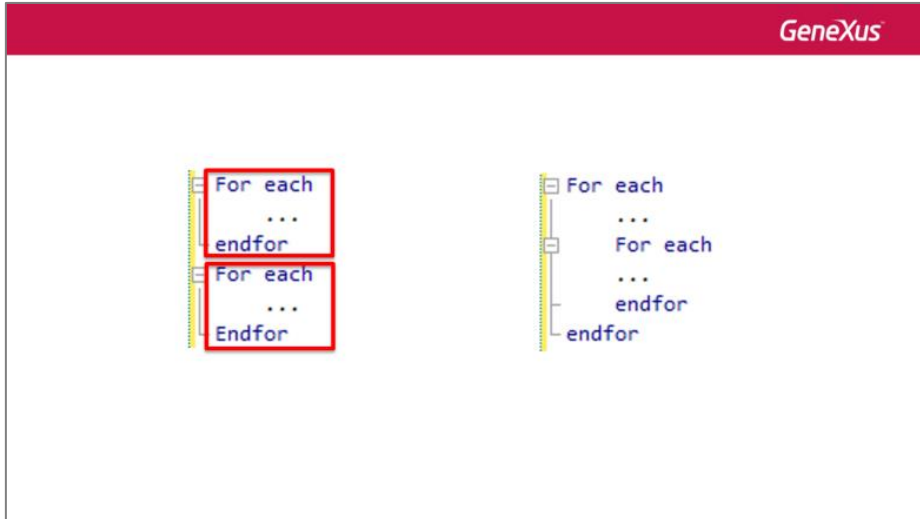
CategoryId is Instantiated

Es por eso que escribimos un For Each que navega las atracciones, sin agregarle una cláusula **where** para filtrar aquellas atracciones que cumplan con la condición de que su categoría coincida con la categoría en la que estamos posicionados dentro del primer for each.

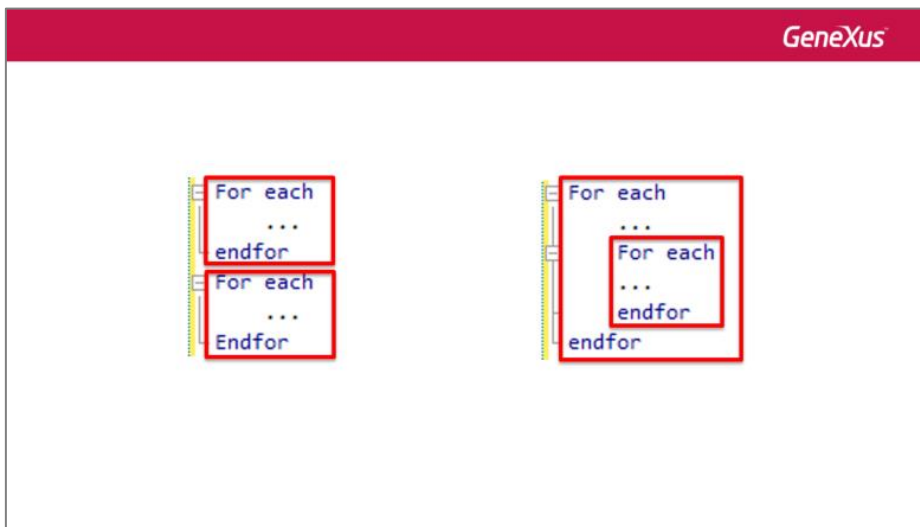
```
where CategoryId = CategoryId
from Category from Attraction
```

¿Cómo hizo GeneXus para establecer ese filtro sin que tuviéramos que escribirlo?  
La respuesta está en la forma en que escribimos los For Each

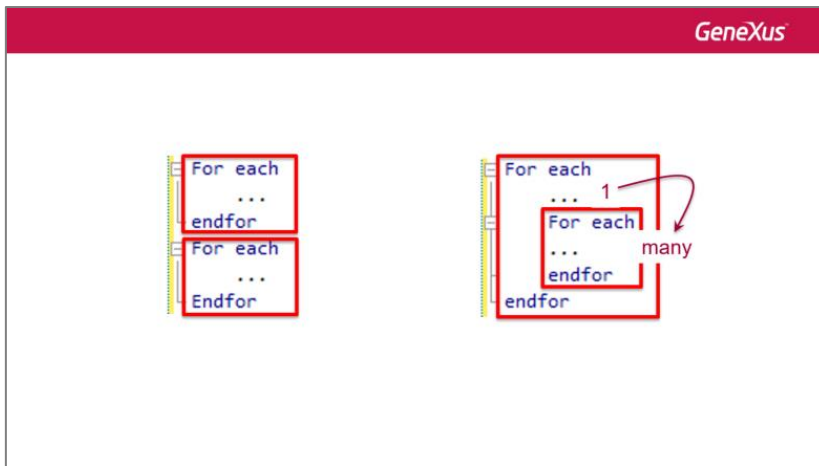
Si dos comandos For each se escriben uno a continuación del otro, entonces **son independientes entre sí**



En cambio si escribimos un For each **dentro** de otro



es porque para cada registro de la primera navegación, queremos recorrer un conjunto de registros en la segunda.



Cuando escribimos comandos For Each anidados

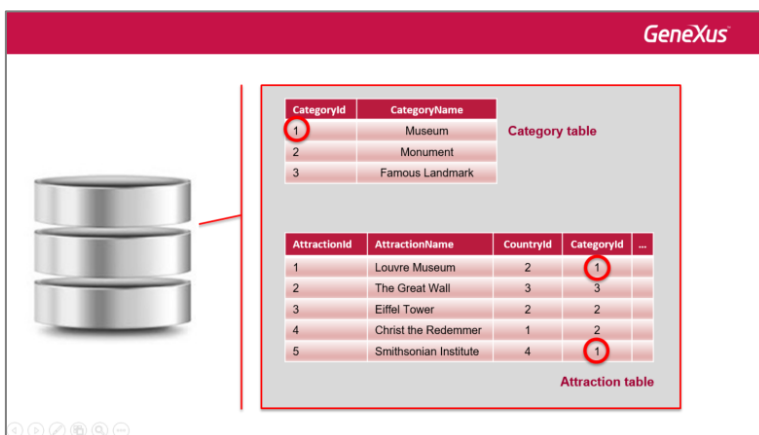
```

1 Print Title
2 For each Category
3   Print Categories
4   Print AttractionColumnTitles
5   For each Attraction
6     Print Attractions
7   Endfor
8 Endfor
9
10

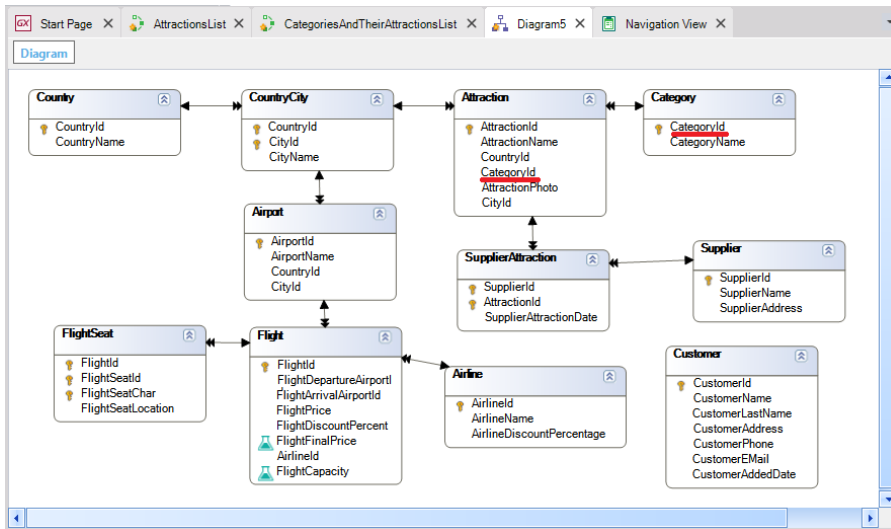
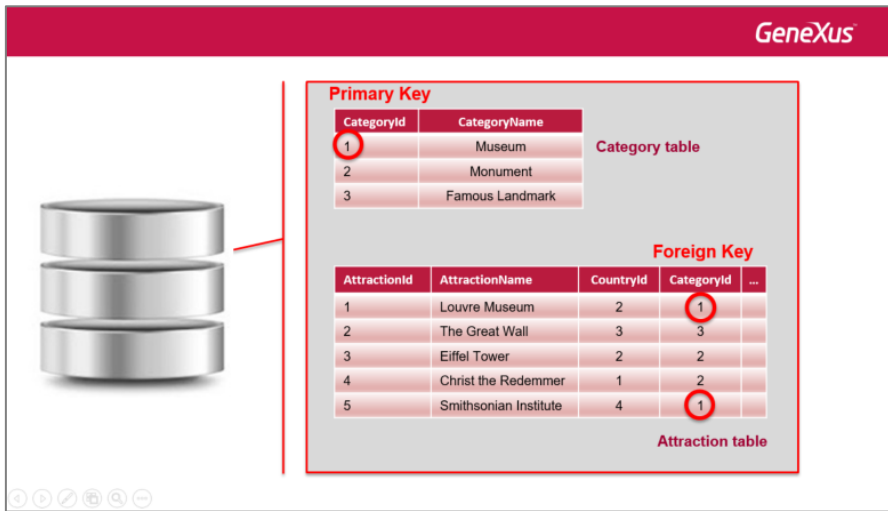
```

GeneXus determina para cada For each, la tabla base que navegará... **y luego busca si existe alguna relación entre esa información.**

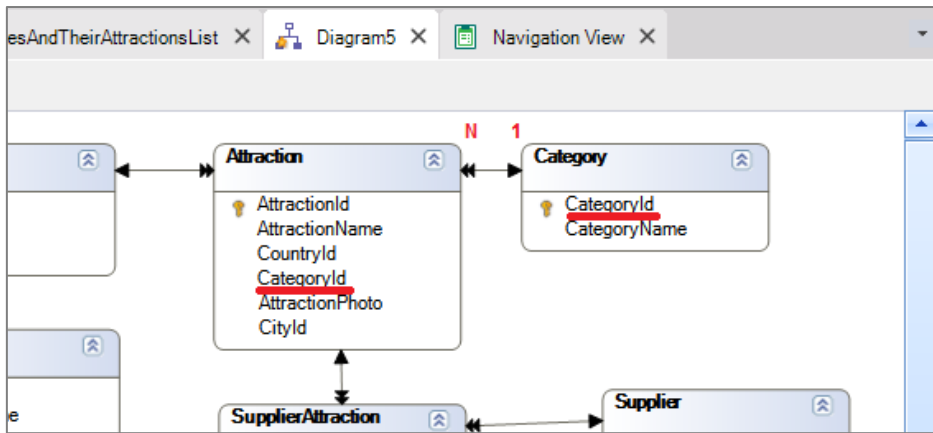
En nuestro caso, la tabla base del For each externo es CATEGORY y la tabla base del For each interno es ATTRACTION



y GeneXus tiene el conocimiento de que hay un atributo en común entre ambas tablas: Este atributo común es **CategoryId**, que es llave primaria en **CATEGORY** y llave foránea en **ATTRACTION**



De modo que el atributo CategoryId hace que se relacionen las tablas ATTRACTION y CATEGORY, como vemos aquí en el diagrama, constituyendo una relación 1 a N



Es decir, que para cada categoría, hay muchas atracciones relacionadas.

Por lo tanto, para cada categoría navegada en el For Each externo, GeneXus ejecuta el For Each que navega la tabla de atracciones

```

1 print Title
2 For each Category
3   print Categories
4   print ColumnTitles
5   For each Attraction
6     print Attractions
7   endfor
8 endfor
9

```

filtrando solamente aquellas atracciones que cumplan que el valor de su CategoryId coincida con el valor del CategoryId de la categoría en la que estamos posicionados.

Es **exactamente** como si en el For each interno, hubiéramos escrito

```

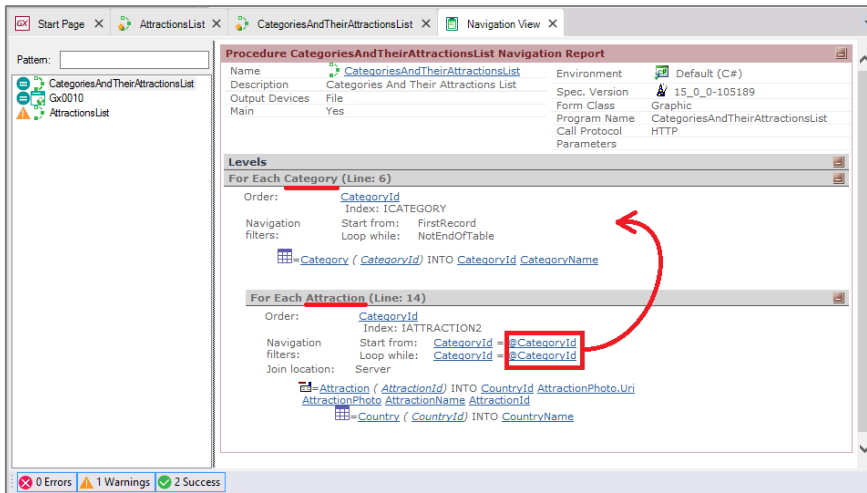
1 print Title
2 For each Category
3   print Categories
4   print ColumnTitles
5   For each Attraction
6     where CategoryId = CategoryId
7     print Attractions
8   endfor
9 endfor

```



Where **CategoryId=CategoryId...** pero no tenemos que ponerlo, porque GeneXus lo detecta y lo aplica.

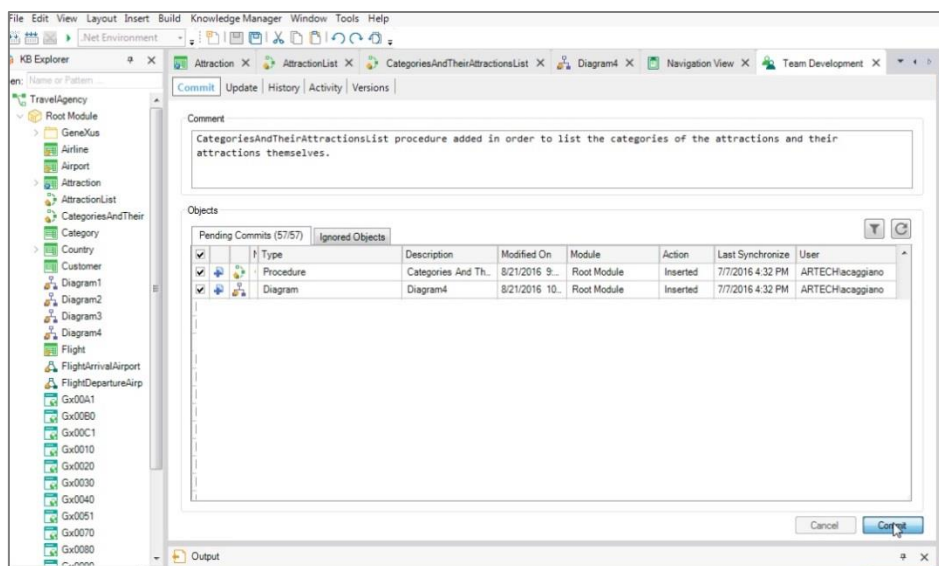
Si vamos al Listado de Navegación de este procedimiento:



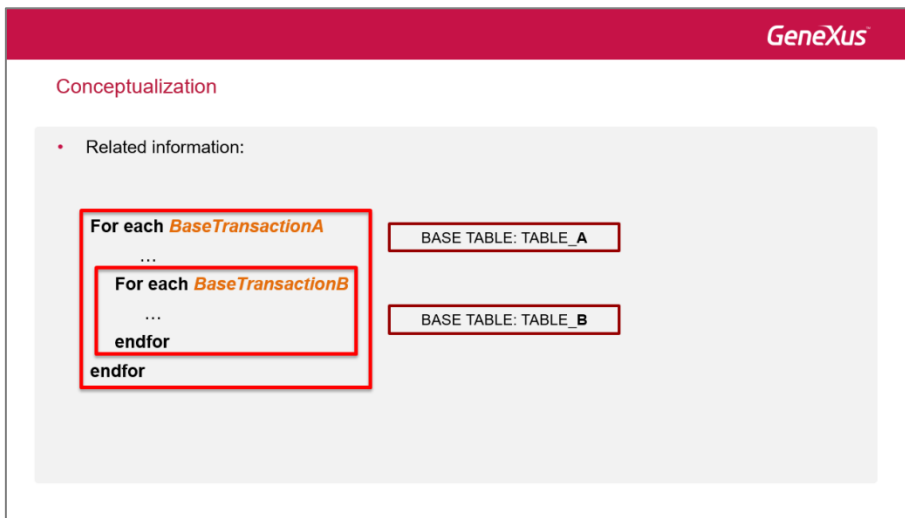
Vemos que nos informa sobre los dos for eachs, y vemos que la tabla base del externo es Category, y la del anidado es Attraction. Además, podemos apreciar que las categorías son recuperadas ordenadas por su identificador, CategoryId y que las atracciones también son ordenadas por ese atributo, pero que en esta tabla es llave foránea, ya que es el atributo que las relaciona, y es por eso que vemos en los filtros de navegación que solamente va a recuperar las atracciones de esta categoría.

Hemos visto qué fácil es obtener información y desplegarla en un listado... pero los procedimientos pueden hacer mucho más que eso. Y lo veremos más adelante.

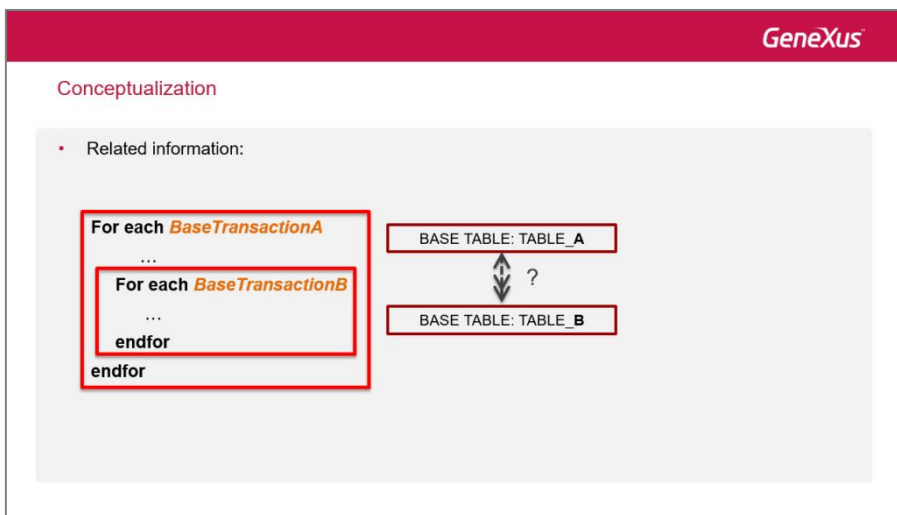
Finalmente actualizamos los cambios en GeneXus Server.



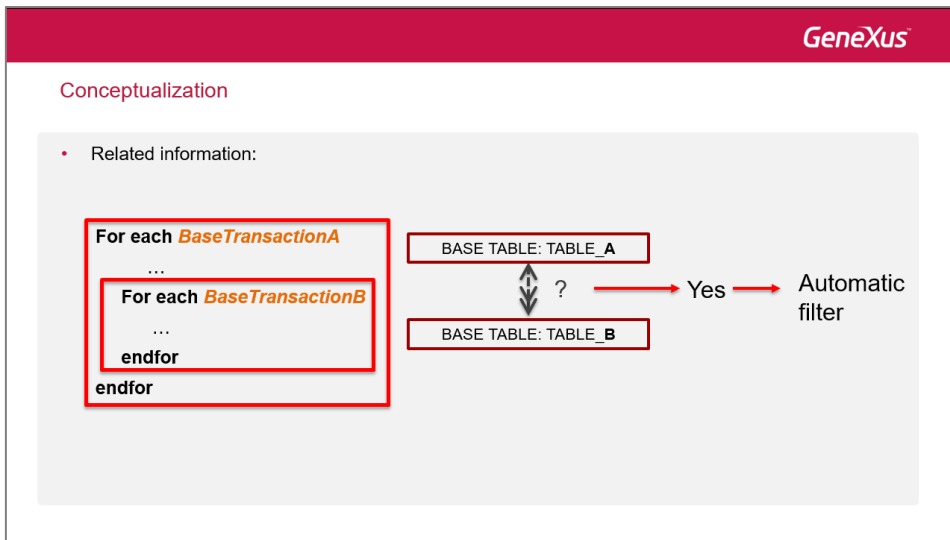
A modo de repaso entonces recordemos que cuando escribimos comandos For Each anidados, GeneXus determina para cada For each la tabla base que navegará



y luego busca si existe alguna relación entre dichas tablas base.

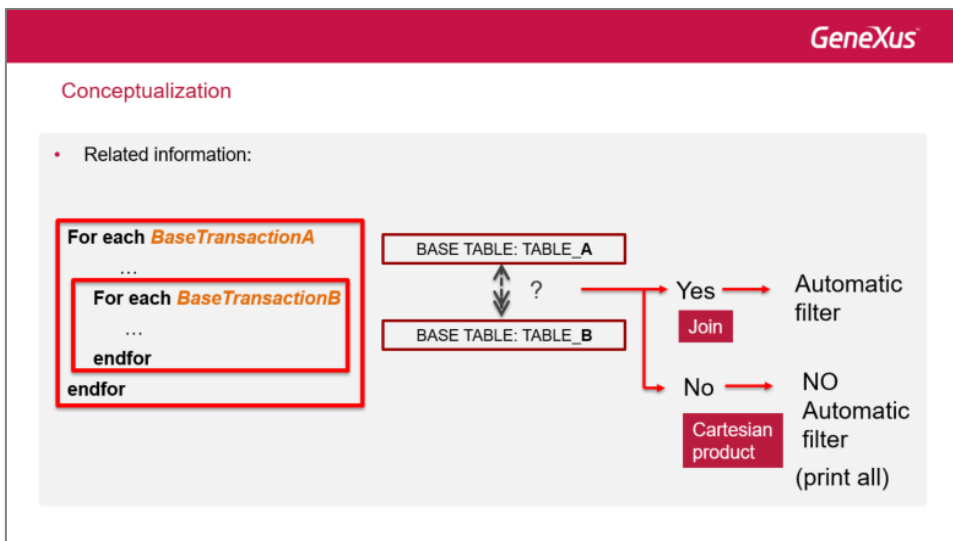


Si la respuesta es sí, como vimos en el listado de este video, entonces aplicará un filtro automático a los registros recorridos por el for each anidado:



A este caso de for eachs anidados donde se filtra la información en base a un criterio de relación, le llamamos **Join**.

Si en cambio la respuesta a la pregunta sobre si existe relación fuera No:



Entonces no aplicará filtro alguno. Se imprimirán todos los registros del for each anidado por cada registro del for each externo. A este tipo de for eachs anidados en los que no se encuentra relación implícita, le llamamos **Producto Cartesiano**. Por supuesto, el desarrollador siempre puede especificar condiciones de filtro explícitas, escribiéndolas directamente en el for each con cláusulas where.

En estos casos supusimos que las tablas eran distintas. En el video siguiente veremos qué ocurre cuando las tablas del for each externo y del anidado son la misma tabla.

**GeneXus™**

Videos	<a href="http://training.genexus.com">training.genexus.com</a>
Documentation	<a href="http://wiki.genexus.com">wiki.genexus.com</a>
Certifications	<a href="http://training.genexus.com/certifications">training.genexus.com/certifications</a>