






Cómo listar información relacionada

Comandos For Eachs anidados

GeneXus[™]

Se pide el siguiente listado:

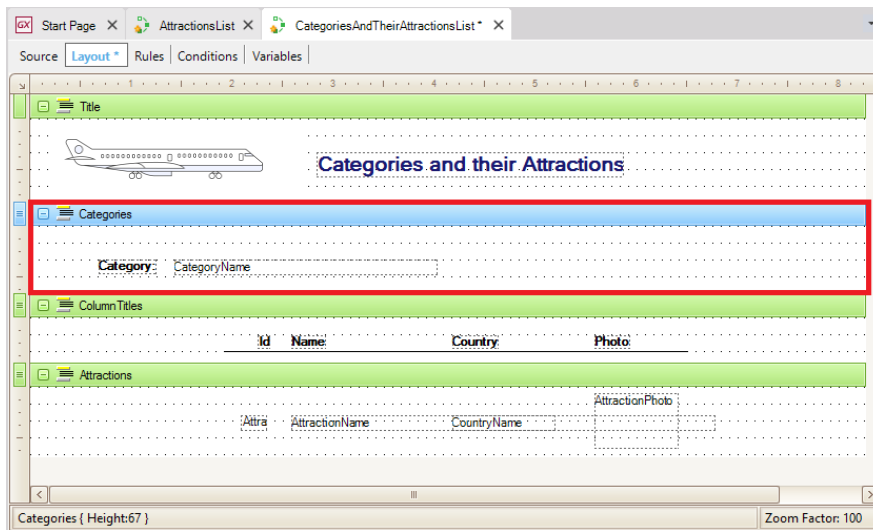
Title
Categories
ColumnTitles
Attractions

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	

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.

Observemos que la gran diferencia con respecto al listado de atracciones que habíamos implementado antes, es que ahora queremos agruparlas por categoría.

Layout



En el Layout agregaremos un printblock a los que ya teníamos.

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.

Source

For each

CategoryId	CategoryName
1	Museum
2	Monument
3	Famous Landmark

For each

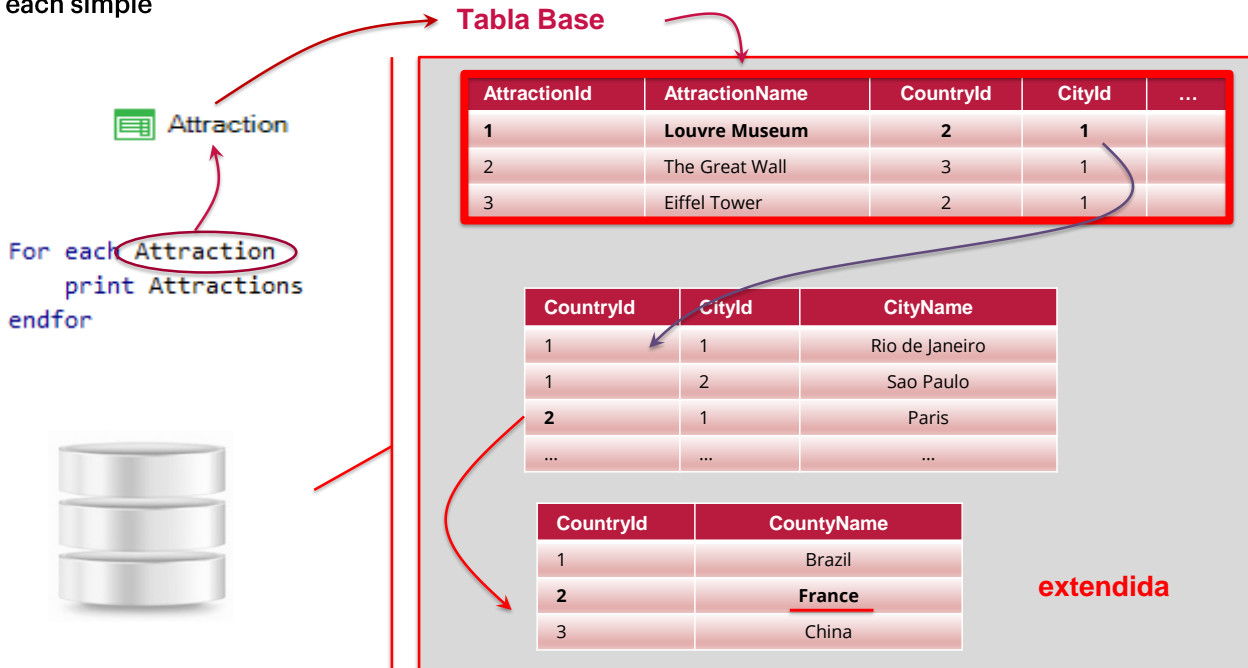
AttractionId	AttractionName	CountryId	CategoryId	...	
→ 1	Louvre Museum	2	1		
	2	The Great Wall	3	3	
	3	Eiffel Tower	2	2	
	4	Christ the Redemmer	1	2	
→ 5	Smithsonian Institute	4	1		

endfor

endfor

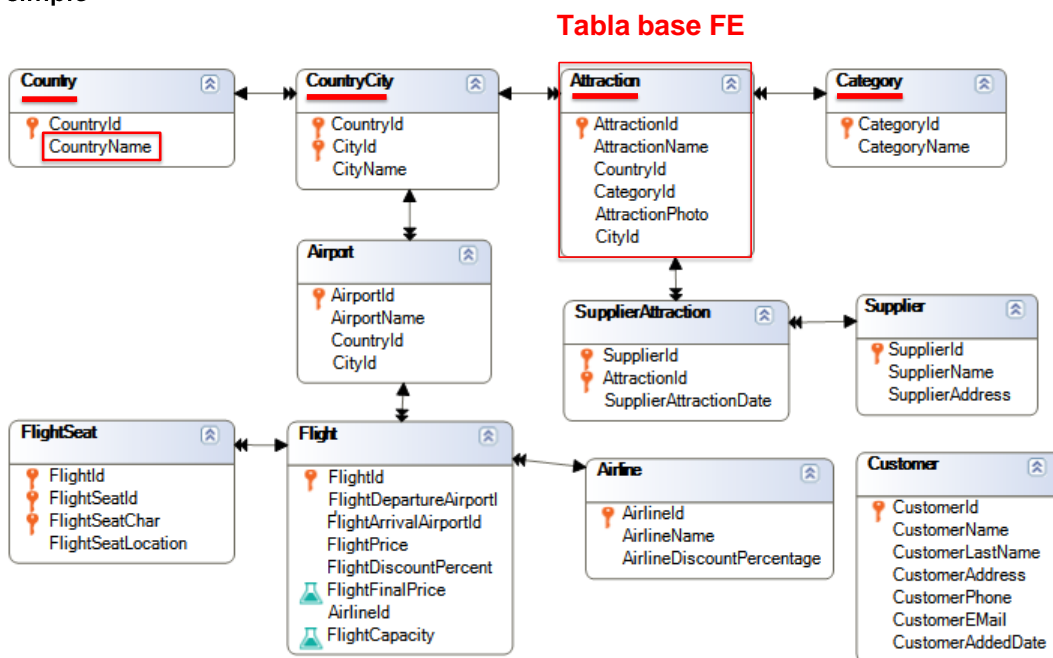
Como tenemos que navegar las categorías y para cada una de ellas **navegar varias** atracciones (las que pertenecen a dicha categoría) este listado es diferente del que hemos resuelto anteriormente.

ANTES:
For each simple



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.

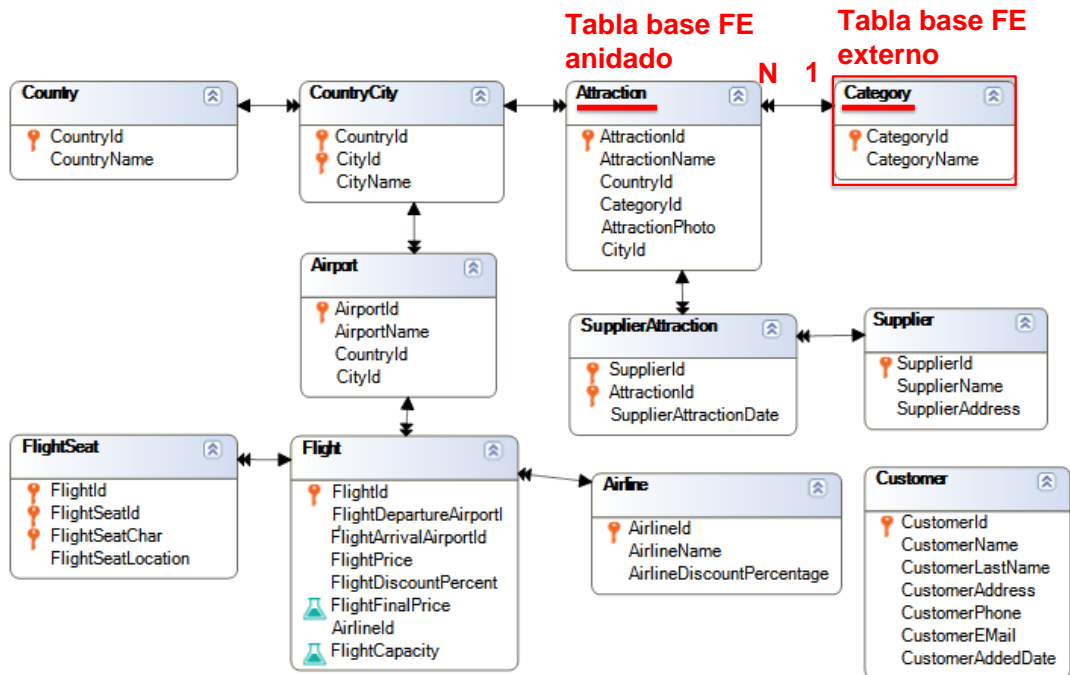
ANTES:
For each simple



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.

AHORA:

For eachs anidados



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.

Diseñando el Source

Source editor content:

```

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

```

Annotations:

- Red box around lines 3-6: **¿Tienen alguna relación?**
- Red box around lines 5-6: **¿Tienen alguna relación?**
- Red arrow from line 3 to **TABLA BASE: CATEGORY**
- Red arrow from line 5 to **TABLA BASE: ATTRACTION**
- Red arrow from line 5 to text: **Aquí estamos posicionados en una categoría (está instanciada)**

ER Diagram:

- Country**: CountryId, CountryName
- CountryCity**: CountryId, CityId, CityName
- Attraction**: AttractionId, AttractionName, CountryId, CategoryId, AttractionPhoto, CityId
- Category**: CategoryId, CategoryName
- Relationships: Country <-> CountryCity <-> Attraction <-> Category. Attraction <-> Category (N:1)

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

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 ColumnTitles**

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.

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

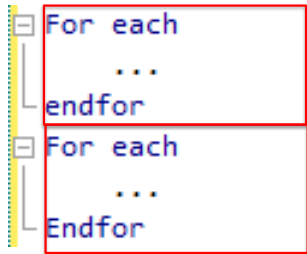
Observemos los `for each`s. Sabemos que un `for each` recorre N registros de una tabla y por cada uno de ellos, ejecuta una serie de instrucciones; las que se encuentran dentro del `for each` Dentro de ese “cuerpo” del primer `for each`, cada vez estaremos posicionados en una y solo una categoría. Decimos 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.

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

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`.

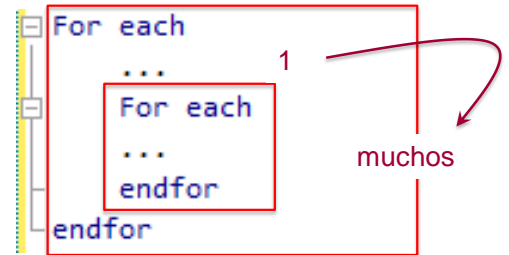
Filtro implícito en For each anidados

For eachs paralelos



Navegaciones
independientes

For eachs anidados



Navegaciones
¿relacionadas?

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

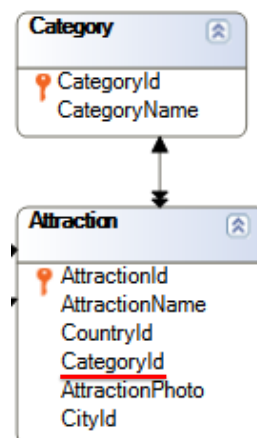
Filtro implícito en For each anidados

Primary Key

CategoryId	CategoryName
1	Museum
2	Monument
3	Famous Landmark

Foreign Key

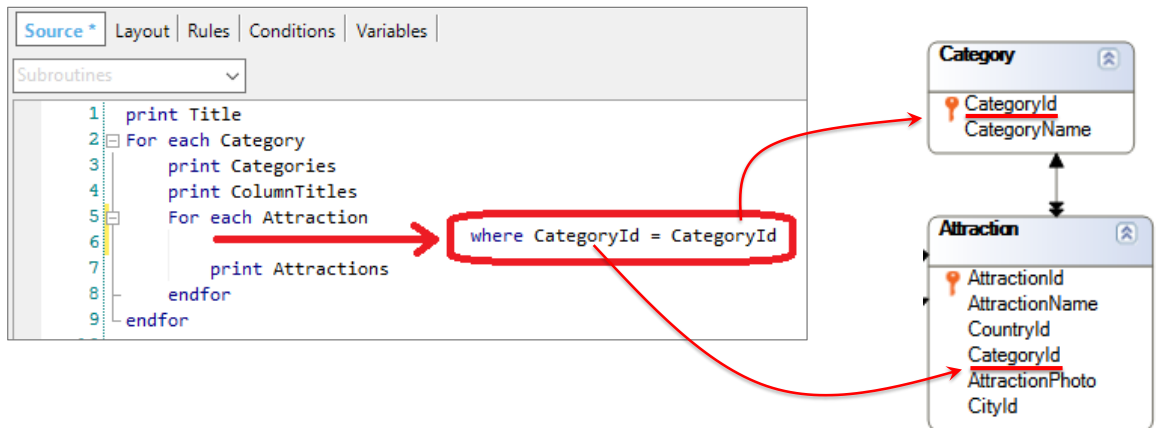
AttractionId	AttractionName	CountryId	CategoryId	...
1	Louvre Museum	2	1	
2	The Great Wall	3	3	
3	Eiffel Tower	2	2	
4	Christ the Redemmer	1	2	
5	Smithsonian Institute	4	1	



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.

Filtro implícito en For each anidados



Por lo tanto, para cada categoría navegada en el For Each externo, GeneXus ejecuta el For Each que navega la tabla de atracciones **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 **Where CategoryId=CategoryId....** pero no tenemos que ponerlo, porque GeneXus lo detecta y lo aplica.

Listado de navegación

Procedure CategoriesAndTheirAttractionsList Navigation Report

Name	CategoriesAndTheirAttractionsList	Environment	Default (C#)
Description	Categories And Their Attractions List	Spec. Version	15_0_0-105189
Output Devices	File	Form Class	Graphic
Main	Yes	Program Name	CategoriesAndTheirAttractionsList
		Call Protocol	HTTP
		Parameters	

Levels

For Each Category (Line: 6)

Order: CategoryId
 Index: ICATEGORY
 Navigation Start from: FirstRecord
 filters: Loop while: NotEndOfTable

For Each Attraction (Line: 14)

Order: CategoryId
 Index: IATTRACTION2
 Navigation Start from: CategoryId = @CategoryId
 filters: Loop while: CategoryId = @CategoryId
 Join location: Server

Table Definitions:

- Attraction (AttractionId) INTO CountryId AttractionPhoto.Uri AttractionPhoto AttractionName AttractionId
- Country (CountryId) INTO CountryName

0 Errors 1 Warning 2 Success

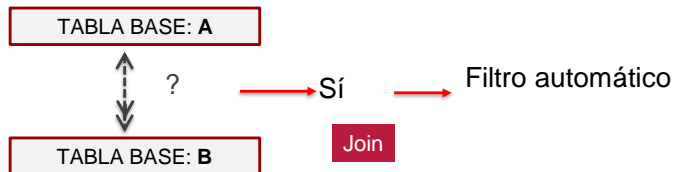
Si vamos al Listado de Navegación de este procedimiento, vemos que nos informa sobre los dos for eachs, y vemos que la tabla base del externo es Category, y la del anidado es Attraction. Además, podemos apreciar que las categorías son recuperadas ordenadas por su identificador, CategoryId y que las atracciones 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.

Repaso

- Información relacionada

```
For each BaseTransactionA  
  ...  
  For each BaseTransactionB  
    ...  
  endfor  
endfor
```



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 esta clase, 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**.

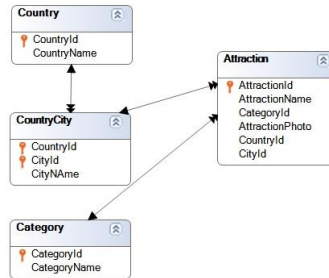
Repaso

Join

```

For each Country.City
  Print countrycity
  ↑ 1
  ↓ N
For each Attraction
  Print attraction
Endfor

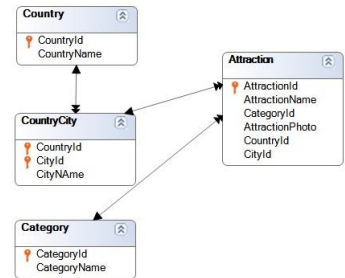
```

**1-N directa****1-N indirecta**

```

for each Country
  print country
  ↑ 1
  ↓ N
For each Attraction
  Print attraction
Endfor
Endfor

```



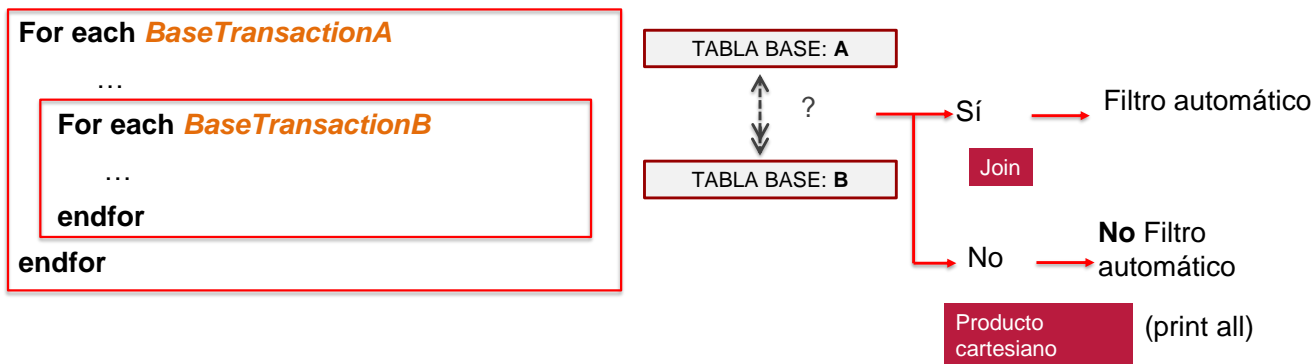
Aquí podemos distinguir dos casos de relación 1-N entre ambas tablas.

La primera es directa. Observemos que las tablas base del for each externo y anidado son CountryCity y Attraction, respectivamente, que están relacionadas por una relación 1 a N.

La segunda es indirecta. Las tablas base del for each externo y anidado son Country y Attraction, que no tienen una relación directa 1 a N, pero sí indirecta, a través de la tabla CountryCity. Dicho de otro modo: observemos que la tabla base del primer for each (Country), está incluida en la tabla extendida de la tabla base del for each anidado (Attraction).

Repaso

- Información relacionada



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. A continuación veremos qué ocurre cuando las tablas del for each externo y del anidado son la misma tabla.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications