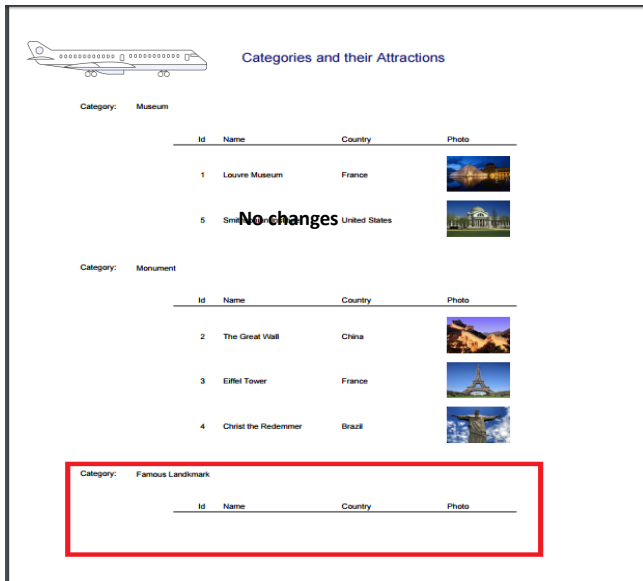


Cómo listar información agrupada

For Eachs anidados



GeneXus

Listando información agrupada






Categories and their Attractions

Category: Museum

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

No changes

Category: Monument

Id	Name	Country	Photo
2	The Great Wall	China	
3	Eiffel Tower	France	
4	Christ the Redemmer	Brazil	

Category: Famous Landmark

Id	Name	Country	Photo
----	------	---------	-------

- Este era el listado de categorías con sus atracciones.
- Famous Landmark no tiene atracciones por el momento.
- ¿Si ahora no se desea que se listen TODAS las categorías, sino sólo aquellas que tengan atracciones?

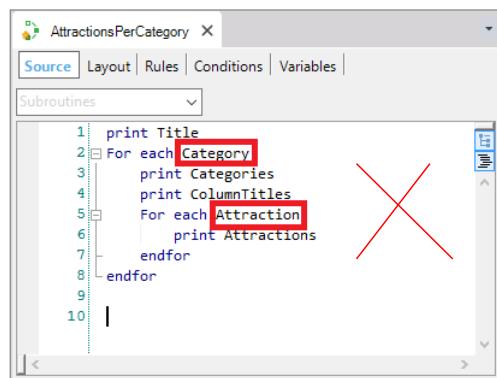
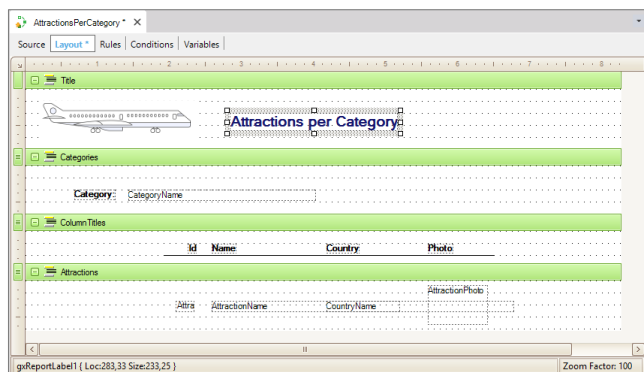
En la sección anterior hemos visto un listado solicitado por la agencia de viajes, que mostraba todas las Categorías de atracciones turísticas y para cada categoría, la lista de atracciones que había ingresadas.

Cambiémosle a la atracción “The Great Wall” su categoría, de modo que “Famous Landmark” ya no tenga ninguna atracción asociada.

Ahora ejecutemos nuevamente el listado. Como podemos observar, este listado nos muestra TODAS las categorías ingresadas, incluso aquellas que no tienen atracciones asociadas.

Si eso no es lo que se desea, es decir, si lo que se quiere es mostrar solamente las categorías con atracciones, ¿cómo hacemos?

¿Por qué no nos sirve solución anterior?



CategoryId	CategoryName
1	Museum
2	Monument
3	Famous Landmark

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

3?

Vamos a implementar esto en otro procedimiento. Para ello salvemos el que tenemos con otro nombre. Y cambiémosle el título del text block.

Si analizamos los For Eachs que teníamos implementados, vemos que la tabla base del For Each externo es Category y la tabla base del For Each anidado es Attraction.

Pero de este modo, como primero se accede a la tabla Category, se imprime la información del registro y recién después se pasa a ejecutar el For each anidado, se estará imprimiendo la categoría antes de saber si tiene atracciones relacionadas o no.

Esto no nos sirve. Necesitamos acceder a las categorías de las atracciones, porque es la única manera de estar seguros de que la categoría que se va a imprimir tiene por lo menos una atracción.

Solución

Foreign Key					Primary Key	
AttractionId	AttractionName	CountryId	CategoryId	...	CategoryId	CategoryName
→ 1	Louvre Museum	2	1		1	Museum
→ 5	Smithsonian Institute	4	1		2	Monument
→ 2	The Great Wall	3	2		3	Famous Landmark
→ 3	Eiffel Tower	2	2			
→ 4	Christ the Redemmer	1	2			

Category: Museum

1 Louvre Museum France



5 Smithsonian Institute United States



Category: Monument

2 The Great Wall China



3 Eiffel Tower France



```

For each Attraction order CategoryId
  print Categories
  For each Attraction
    print Attractions
  endfor
endfor

```

→ Criterio de agrupamiento o "corte"

Corte de control

La idea será agrupar las atracciones de la tabla Attraction por categoría, y luego recorrer esos grupos, imprimiendo para cada uno la categoría (para la que habrá que acceder a la tabla Category para recuperar su nombre); y también imprimiendo cada atracción del grupo.

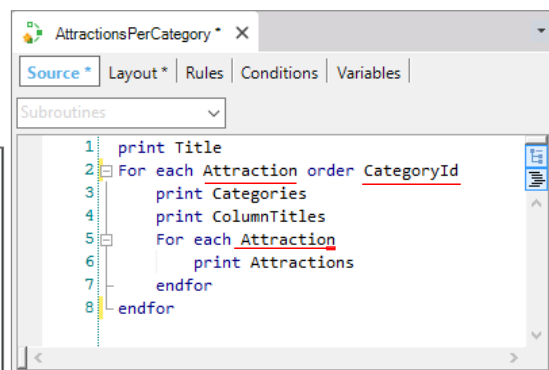
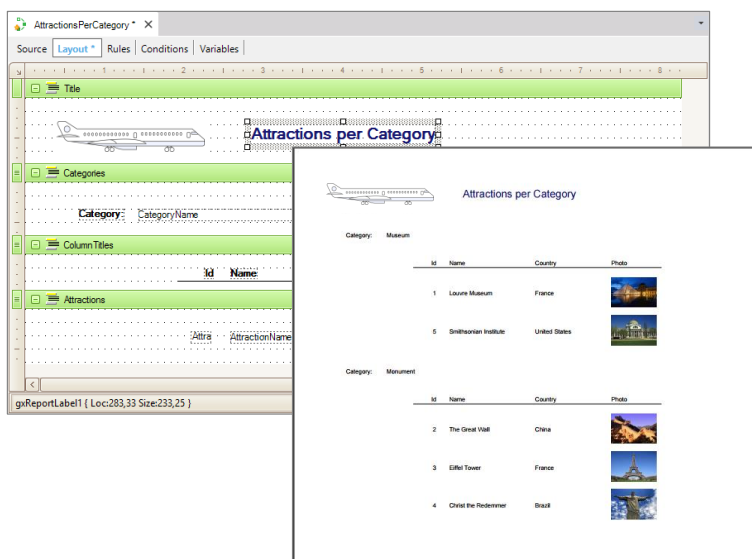
Para luego pasar al siguiente grupo... y así sucesivamente.

En definitiva, necesitamos recorrer únicamente la tabla Attraction, primero agrupándola por categoría e imprimiendo la categoría; y luego imprimiendo de cada grupo de categoría (navegando la misma tabla Attractions, claro) sus atracciones.

Observemos que la forma de indicarle a GeneXus que se quiere agrupar por CategoryId, es especificando la cláusula Order.

Este caso de for eachs anidados que recorren la misma tabla se conoce como **corte de control**.

Listando información agrupada: corte de control



Hagamos los cambios en nuestro procedimiento...

Primero cambiamos la transacción base del for each externo, por Attraction...

Y agregamos la cláusula Order, para ordenar por CategoryId, que en el caso del corte de control será además para otra cosa más fuerte: para **agrupar** por ese atributo.

Ejecutemos. Vemos que efectivamente no está siendo listada la categoría "Famous Landmark" que no tiene atracciones.

Listando información agrupada: corte de control

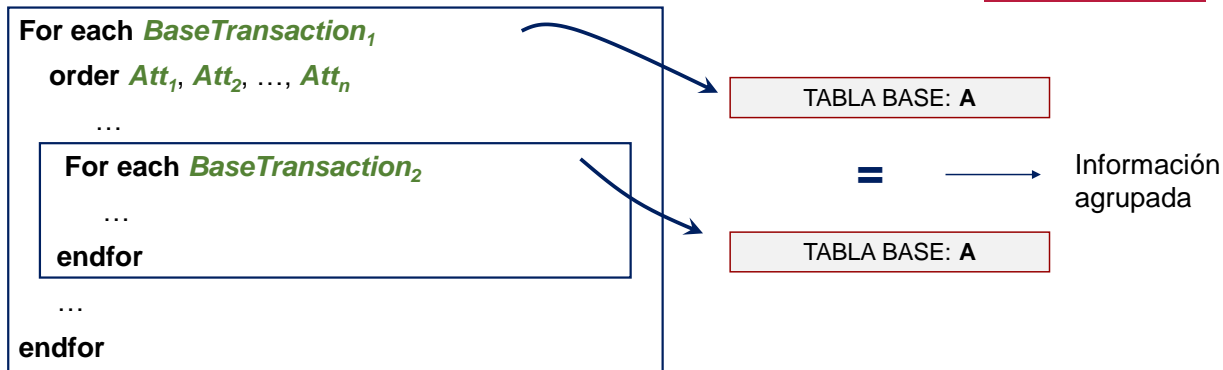
The screenshot displays the 'Navigation Report' for the procedure 'AttractionsPerCategory'. The report is organized into several sections:

- Procedure Information:**
 - Name: AttractionsPerCategory
 - Description: Attractions Per Category
 - Output Devices: File
 - Main: Yes
 - Environment: Default (C#)
 - Spec. Version: 15_0_0-105189
 - Form Class: Graphic
 - Program Name: AttractionsPerCategory
 - Call Protocol: HTTP
 - Parameters:
- Levels:**
 - For Each Attraction (Line: 6):**
 - Order: CategoryId
 - Index: IATTRACTION2
 - Navigation: Start from: FirstRecord
 - Filters: Loop while: NotEndOfTable
 - Join location: Server
 - Structure:
 - =Attraction (AttractionId)
 - =Country (CountryId)
 - =Category (CategoryId)
 - Break Attraction (Line: 14):**
 - Order: CategoryId
 - Index: IATTRACTION2
 - Navigation: Loop while: CategoryId = @CategoryId
 - Filters:
 - Join location: Server
 - Structure:
 - =Attraction (AttractionId)
 - =Country (CountryId)
- Status Bar:** 0 Errors, 0 Warnings, 1 Success

Si observamos el listado de navegación resultante, podemos ver que nos informa de un for each a la tabla Attraction, ordenado por CategoryId... y que será cortado, Break, por el for each anidado sobre la misma tabla, Attraction. Observemos que en este “corte” se recorren solamente las atracciones de la categoría del grupo.

Conceptualización

- Cortar la información por algún criterio

✓ **Misma tabla base****Corte de control**

Conceptualicemos, entonces, cómo se define un corte de control.

Se trata de For eachs anidados... Con la misma tabla base para todos los For Eachs...

Conceptualización

- Cortar la información por algún criterio

✓ **Criterio de corte:
cláusula order**

For each *BaseTransaction*₁

order *Att*₁, *Att*₂, ..., *Att*_n

...

For each *BaseTransaction*₂

...

endfor

...

endfor

Criterio de corte

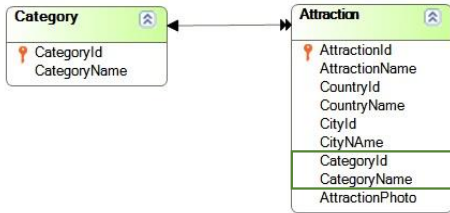
TABLA BASE: A

Agrupada por...

Y la cláusula Order para establecer el criterio de corte.

Otro ejemplo: Cláusula Unique

Listar solamente aquellas categorías que tienen atracciones turísticas registradas. No se desean ver nombres de categorías repetidos.



```

Print Title
For each Attraction
  Unique CategoryName
  Print Categories
Endfor
  
```

Tabla base: **ATTRACTION**

Analicemos ahora otra situación.

Supongamos que necesitamos listar solamente aquellas categorías que tienen atracciones turísticas registradas. ¿Cómo podemos resolverlo?

Si miramos el diseño de transacciones, vemos claramente que las categorías relacionadas con alguna atracción turística son las que se encuentran en Attraction, como clave foránea.

Lo primero que podemos pensar es en un For each con Attraction como transacción base y listar luego el nombre de la Categoría. Y si bien este listado efectivamente nos muestra el nombre de las categorías que tienen alguna atracción asociada, esos nombres se repiten, ya que por ejemplo, existen varias atracciones que son Monumentos.

¿Cómo podemos controlar que estos nombres listados no se repitan? O sea, ¿que se muestren una sola vez?

Utilizando la cláusula Unique.

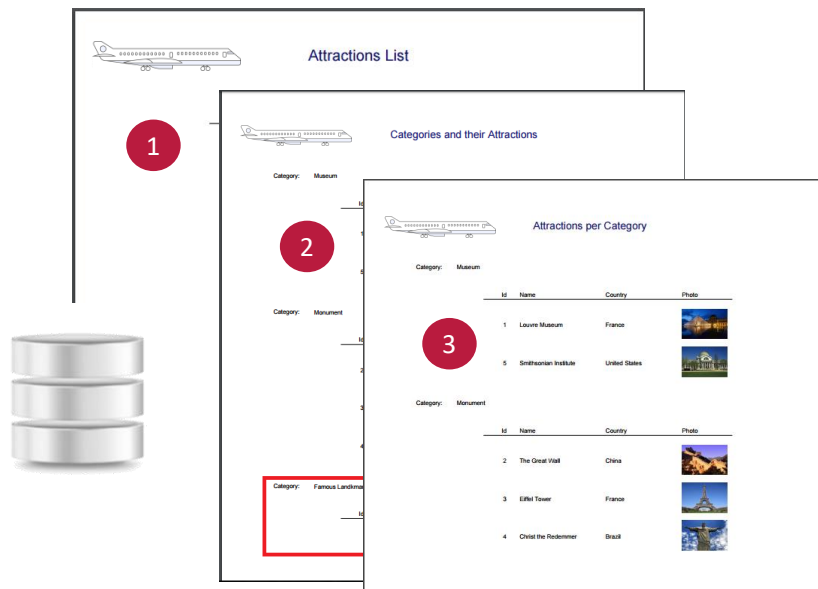
Esta cláusula permite indicar el atributo, o conjunto de atributos, cuyo valor no debe repetirse en la salida de la consulta. Dichos atributos deben pertenecer a la tabla extendida de la tabla base del For each.

Resumen

Resumiendo

- Listados
 - ✓ Layout
 - ✓ Source

comando
For each



Con estos videos hemos visto las facilidades que nos proporciona GeneXus para hacer listados simples (que navegan una única tabla), o más complejos, navegando información de varias tablas relacionadas (join), o de la misma tabla, pero agrupada por algún criterio (corte de control).

El comando que utilizamos en todos los casos para acceder a la base de datos es el **For each**.

Resumiendo

- Listados
 - ✓ Layout
 - ✓ Source



Id	Name	Country	Photo
1	Louvre Museum	France	
2	The Great Wall	China	
3	Eiffel Tower	France	

comando
For each



For each simple

1 For each *BaseTransaction*
 order *Att₁, Att₂, ..., Att_n*
 where *condition₁*
 where *condition₂*
 ...
 where *condition_n*
MainCode
 endfor

En el primer caso, usamos un for each simple, en el que a través de su Transacción Base se infiere la tabla a ser navegada.

Resumiendo

- Listados
- ✓ Layout
- ✓ Source

comando
For each



For eachs anidados

Categories and their Attractions

Category	Museum																
	<table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Country</th> <th>Photo</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Louvre Museum</td> <td>France</td> <td></td> </tr> <tr> <td>5</td> <td>Smithsonian Institute</td> <td>United States</td> <td></td> </tr> </tbody> </table>	ID	Name	Country	Photo	1	Louvre Museum	France		5	Smithsonian Institute	United States					
ID	Name	Country	Photo														
1	Louvre Museum	France															
5	Smithsonian Institute	United States															
	<table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Country</th> <th>Photo</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>The Great Wall</td> <td>China</td> <td></td> </tr> <tr> <td>3</td> <td>Eiffel Tower</td> <td>France</td> <td></td> </tr> <tr> <td>4</td> <td>Christ the Redeemer</td> <td>Brazil</td> <td></td> </tr> </tbody> </table>	ID	Name	Country	Photo	2	The Great Wall	China		3	Eiffel Tower	France		4	Christ the Redeemer	Brazil	
ID	Name	Country	Photo														
2	The Great Wall	China															
3	Eiffel Tower	France															
4	Christ the Redeemer	Brazil															
	<table border="1"> <thead> <tr> <th>ID</th> <th>Name</th> <th>Country</th> <th>Photo</th> </tr> </thead> <tbody> </tbody> </table>	ID	Name	Country	Photo												
ID	Name	Country	Photo														

2

```
For each BaseTransaction
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
```

```
...
For each BaseTransaction
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
...
endfor
...
endfor
```

1
≠
N

Join

En el segundo tenemos un par de for eachs anidados, en el que a partir de transacciones base diferentes, se descubre una relación de uno a muchos entre la información de cada for each.

Resumiendo

- Listados
- ✓ Layout
- ✓ Source

comando
For each



Attractions per Category

Category: Museum			
ID	Name	Country	Photo
1	Louvre Museum	France	
5	Smithsonian Institute	United States	

Category: Monument			
ID	Name	Country	Photo
2	The Great Wall	China	
3	Eiffel Tower	France	
4	Christ the Redeemer	Brazil	

3

3

For each *BaseTransaction*order *Att₁, Att₂, ..., Att_n*where *condition₁*where *condition₂*

...

where *condition_n*

...

For each *BaseTransaction*order *Att₁, Att₂, ..., Att_n*where *condition₁*where *condition₂*

...

where *condition_n*

...

...

endfor

...

endfor

=

For eachs anidados

Corte de control

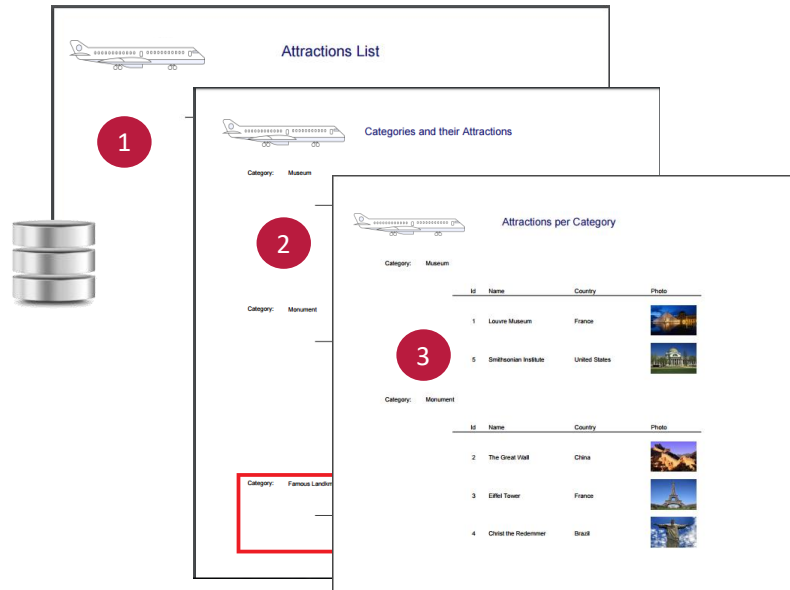
Mientras que en el tercero también tenemos un par de for eachs anidados, pero cuyas transacciones base **coinciden**. GeneXus entiende que se quiere “cortar” o “agrupar” la información de la tabla a ser recorrida, por el atributo o conjunto de atributos que se especifican en la cláusula order del for each exterior.

Y más...

- Listados
- ✓ Layout
- ✓ Source

comando
For each

- Paneles
- Objeto Query



Más adelante veremos otros mecanismos para hacer consultas a la base de datos y obtener información de forma flexible y vistosa.

*GeneXus*TM

training.genexus.com
wiki.genexus.com