

**FOR EACHS**  
Más  
conocimientos  
sobre For eachs  
anidados

CASOS  
Y  
NAVEGACIÓN

- Tablas base  $\neq$

¿ Fe externo



Fe anidado ?

¿Existe relación implícita que lo vincule con un número N de registros del For each anidado?

Sí

**Join:** se recuperan algunos registros del anidado, los relacionados.

(Caso 1)

No

**Producto cartesiano:** se recuperan todos los registros del anidado.

(Caso 2)

- Tablas base =

**Corte de Control:** Corresponde al caso en el que queremos recuperar información por grupos.

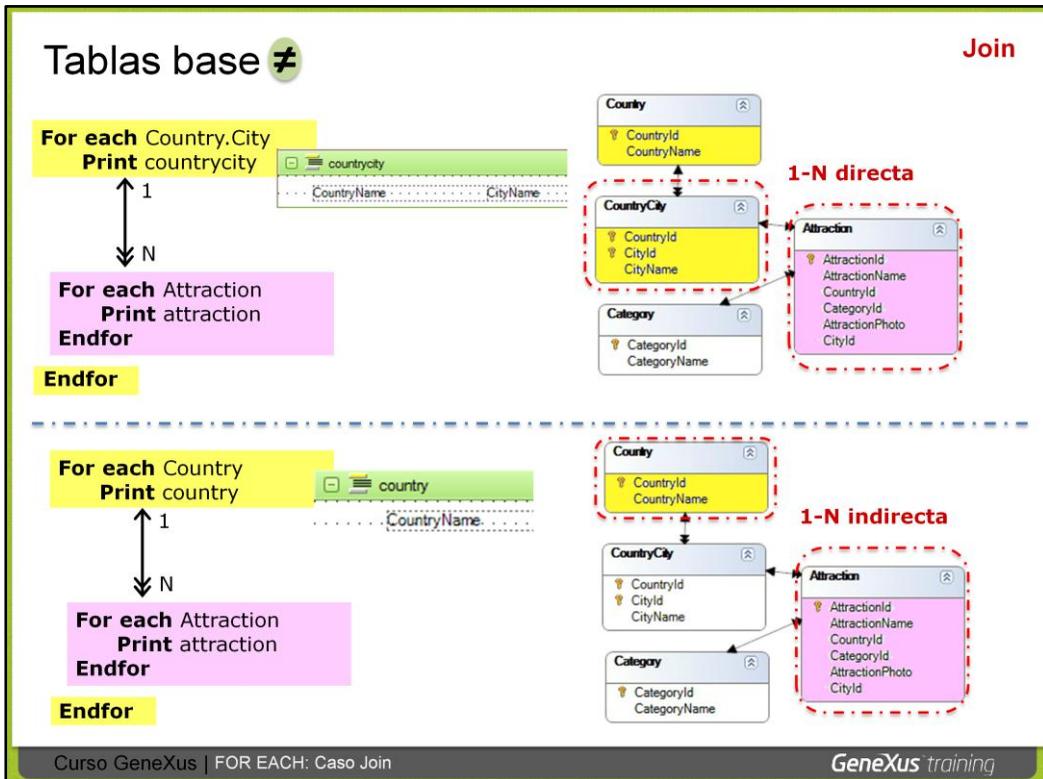
(Caso 3)

De la determinación de las tablas base, surgen los tres casos de For eachs anidados que ya hemos estudiado antes, y que aquí queremos conceptualizar.

Cuando las **tablas base** son **distintas**, se abren dos posibilidades: o existe **relación 1 a N directa o indirecta** entre ellas, o no existe. En el primer caso, por cada registro del for each principal, el for each anidado ejecutará sus instrucciones solamente para los N registros relacionados. A esta operación de cortar la información de una tabla, por la de otra, se la conoce como **Join**.

En el segundo caso, cuando no existe relación, por cada registro considerado en el for each principal, el for each anidado ejecutará sus instrucciones para todos los registros de la otra tabla, dado que no encontró relación entre ellas. La operación se conoce como **Producto Cartesiano**.

Cuando las **tablas base** son **la misma**, se tratará de un caso conocido como **Corte de control**: es cuando necesitamos agrupar la información de una tabla, ejecutar ciertas instrucciones que tienen en cuenta la info común del grupo y luego recorrer cada miembro del mismo, y ejecutar otras instrucciones, para a continuación, pasar al siguiente grupo y repetir el proceso. En este caso es fundamental especificar los atributos que conforman el grupo, mediante la cláusula **order**.



Aquí tenemos dos casos de relación 1 a N.

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).

**Join**

**For Each CountryCity (Line: 2)**

Order: CountryId , CityId  
 Index: ICOUNTRYCITY

Navigation Start from: FirstRecord  
 filters: Loop while: NotEndOfTable  
 Join location: Server

=CountryCity( CountryId , CityId)  
 =Country( CountryId)

---

**For Each Attraction (Line: 5)**

Order: CountryId , CityId  
 Index: IATTRACTION1

Navigation Start from: CountryId = @CountryId  
 filters: CityId = @CityId  
 Loop while: CountryId = @CountryId  
 CityId = @CityId

=Attraction( AttractionId)

**1-N directa**

**1-N indirecta**

---

**For Each Country (Line: 2)**

Order: CountryId  
 Index: ICOUNTRY

Navigation Start from: FirstRecord  
 filters: Loop while: NotEndOfTable

=Country( CountryId)

---

**For Each Attraction (Line: 4)**

Order: CountryId  
 Index: IATTRACTION1

Navigation Start from: CountryId = @CountryId  
 filters: CountryId = @CountryId

=Attraction( AttractionId)

Curso GeneXus | FOR EACH: Caso Join

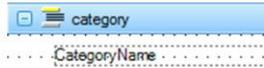
*GeneXus training*

Los listados de navegación indican claramente el Join: para el for each anidado no se recorre toda la tabla. Obsérvese que en lugar de ordenar la recorrida por la clave primaria de ATTRACTION, que es AttractionId, lo hace por el atributo relación, para el que cuenta con índice automáticamente creado, por clave foránea. De esta manera, el acceso a la base de datos estará optimizado.

# Tablas base $\neq$

## Producto Cartesiano

**For each** Category  
**Print** category



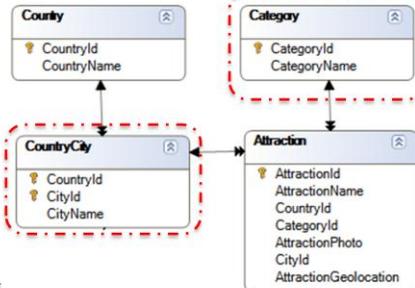
**For each** Country.City  
**Print** countrycity



**Endfor**  
**Endfor**

```
For Each Category (Line: 9)
  Order:      CategoryId
              Index: ICATEGORY
  Navigation Start from: FirstRecord
  filters:    Loop while: NotEndOfTable
   =Category( CategoryId)
```

```
For Each CountryCity (Line: 12)
  Order:      CountryId , CityId
              Index: ICOUNTRYCITY
  Join location: Server
   =CountryCity( CountryId, CityId)
   =Country( CountryId)
```



En este caso GeneXus no logra encontrar una relación 1-N directa o indirecta entre las tablas y por lo tanto no aplica filtros implícitos a los registros del For each anidado, es decir, realiza un producto cartesiano entre las tablas: para cada registro de la tabla base del for each externo (CATEGORY), considera todos los registros de la tabla base del anidado (COUNTRYCITY).

## Tablas base =

**For each** Attraction order CountryName

**Print** country

country
Country: CountryName

**For each** Attraction order CityName

**Print** city

city
City: CityName

**Print** attractionTitles

attractionTitles
Attraction

**For each** Attraction

**Print** attraction

attraction
AttractionName

**Endfor**

**Endfor**

**Endfor**

## Corte de control

Country: Brazil	
City: Rio de Janeiro	Attraction
	The Christ Redemptor
City: Sao Paulo	Attraction
	Museum of Football
Country: China	
City: Beijing	Attraction
	Great Wall
Country: Egypt	
City: Cairo	Attraction
	Egypt Pyramids
Country: France	
City: Niza	Attraction
	Musee Mollat
	Musee National Marc Chagall
City: Paris	Attraction
	Louvre Museum
	Eiffel Tower
Country: United States	
City: Washington	Attraction
	The Smithsonian Institute

Aquí vemos un caso en que se estamos queriendo listar cada país, y para él cada ciudad, y para ella cada atracción. La restricción: queremos hacerlo sólo para los países y ciudades para los que existen atracciones turísticas registradas.

Es decir, tendremos que implementar un **corte de control doble**: donde primero agrupemos por país, y dentro de ese grupo, agrupemos luego por ciudad, y dentro de este último, mostremos los nombres de todas las atracciones. Para ello:

1. Todas las tablas bases deben ser ATTRACTION
2. Tendremos que marcar los criterios de agrupamiento utilizando las **cláusulas order**. Recordemos que para un corte de control, el order tiene un peso muy fuerte: no sólo está marcando por qué atributo o atributos listar la información, sino que está especificando cómo ésta se va a agrupar.

Podríamos especificar un order para el for each más interno, pero ese order sí tendrá únicamente su uso convencional. Es decir, ese sí será utilizado únicamente para ordenar.

# Tablas base =

Corte de control

For each Attraction order CountryName

Print country  country  
Country: CountryName

For each Attraction order CityName

Print city  city  
City: CityName

Print attractionTitles  attractionTitles  
Attraction

For each Attraction

Print attraction  attraction  
AttractionName

Endfor

Endfor

Endfor

CountryId	CityId	AttractionName	Attractionid	Attracti...	Categoryid	Attr...
1	1	The Christ Redeemer ...	4	<Binary ...	2	-21.2...
1	2	Museum of Football ...	9	<Binary ...	1	...
2	1	Louvre Museum ...	1	<Binary ...	1	48.86...
2	1	Eiffel Tower ...	3	<Binary ...	2	48.85...
2	2	Musee Matisse ...	7	<Binary ...	1	43.72...
2	2	Musee National Marc Chagall ...	8	<Binary ...	1	43.70...
3	1	Great Wall ...	2	<Binary ...	3	42.68...
4	1	The Smithsonian Institute ...	5	<Binary ...	1	38.95...
5	1	Egypt Pyramides ...	6	<Binary ...	3	30.05...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Tenemos un doble corte de control, lo que implica tres for eachs. En el orden del primero se establece el grupo más externo, en el del segundo, el grupo interior.

En el ejemplo, por simplicidad, mostramos estos agrupamientos para un contenido dado de la tabla ATTRACTION, si en lugar de ordenar por CountryName primero y por CityName segundo, lo hubiéramos hecho por sus identificadores (es decir: CountryId y CityId). El caso del país 2 que tiene atracciones para un par de ciudades, dejará más claro el funcionamiento del corte de control.

El listado de navegación nos informará sobre esto...

## Tablas base =

Corte de control

```
For Each Attraction (Line: 17)
  Order: CountryName , CityName
  Navigation Start from: FirstRecord
  filters: Loop while: NotEndOfTable
  Join location: Server
  =Attraction( AttractionId)
  =Country( CountryId)
  ~CountryCity( CountryId, CityId)

Break Attraction (Line: 21)
  Order: CountryName , CityName
  Navigation Loop while: CountryName = @CountryName
  filters:
  Join location: Server
  =Attraction( AttractionId)
  =Country( CountryId)
  ~CountryCity( CountryId, CityId)

Break Attraction (Line: 23)
  Order: CountryName , CityName
  Navigation Loop while: CountryName = @CountryName and CityName = @CityName
  filters:
  Join location: Server
  =Attraction( AttractionId)
  =Country( CountryId)
  ~CountryCity( CountryId, CityId)
```

Vemos la palabra Break para cada for each interno, indicando la misma tabla base, ATTRACTION, y por tanto, un corte de control.

Además, recorrerá esa tabla base una única vez, para lo cual necesita ordenar por la concatenación de los atributos que aparezcan en los orders de los for eachs. Es por eso que elige CountryName, CityName.

Obsérvese que en el segundo for each corta por país, iterando sobre el país en el que se encuentra posicionado en el primer for each, y el tercer for each, corta por ciudad, iterando sobre la ciudad en la que se encuentra posicionado en el segundo for each.

Piense cuál será la ejecución del listado anterior, si en lugar de haber ordenado el primer for each por CountryName y el segundo por CityName, hubiéramos ordenado por el par CountryName, CityName. El efecto no es el mismo de lo que necesitábamos. Observe que en ese caso el listado de navegación diferirá de este que ve arriba, en el segundo for each. Loop while allí dirá "CountryName = @CountryName **and** CityName = @CityName".

*GeneXus<sup>®</sup> training*  
*training.genexus.com*