Más sobre
For Eachs anidados
Casos y navegación

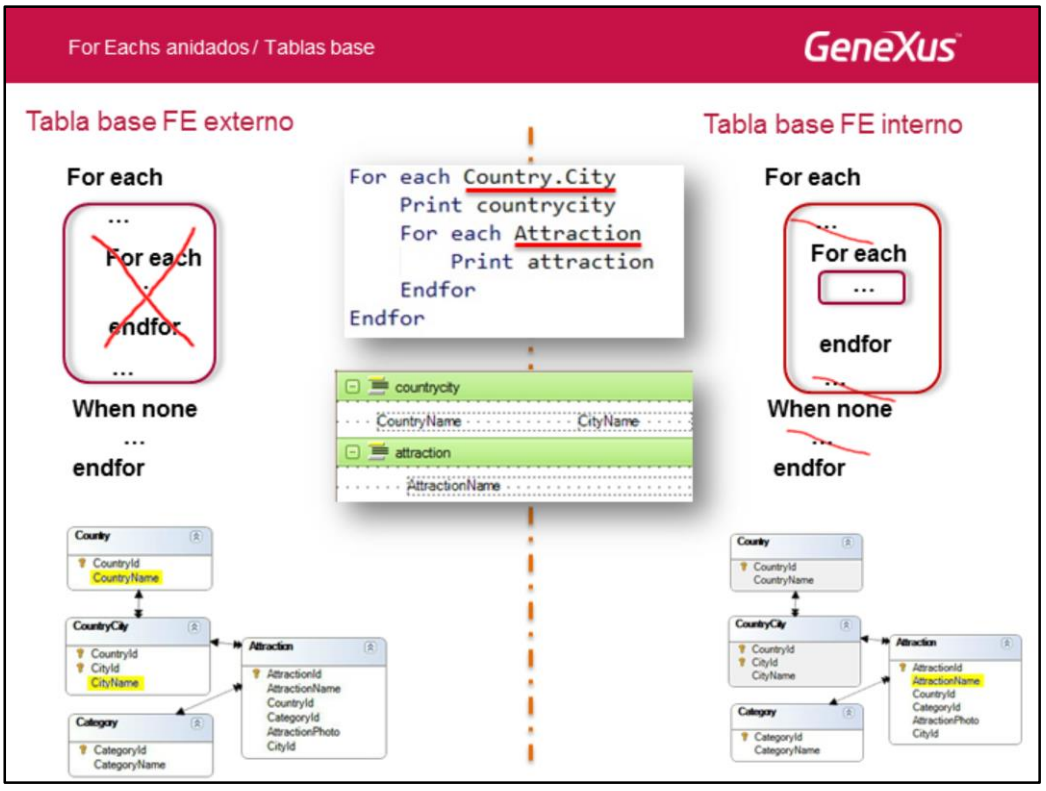
GeneXus 15

Tabla base

Tablas base y Navegación

- Lo primero que debe hacer GeneXus ante un caso de for eachs anidados es determinar las tablas base de cada uno.
  - Cuando el desarrollador indica Transacción Base, esto es inmediato.
  - Cuando no lo hace, GeneXus debe determinar cada tabla base a partir de los atributos presentes en cada for each. Este caso, más complejo, no será abordado en este curso.

- Luego, a partir de allí, define la navegación para resolver la consulta múltiple. Se dará uno de tres casos:
  - Join
  - Producto Cartesiano
  - Corte de control



Como decíamos, lo primero que hace GeneXus al encontrar un par de for eachs anidados es determinar la tabla base de cada uno, en forma ordenada, de afuera hacia adentro, empezando por el más externo. Recién luego determina la navegación.

Por cada for each interviene la transacción base indicada, y únicamente los atributos propios de ese for each: tanto del orden, where, etc., así como los que estén en su cuerpo, exceptuando los que se encuentren dentro de un for each anidado. Es decir, quitando el for each anidado, se determina la tabla base como en el caso de un for each simple. No se toman en cuenta nunca los atributos de la cláusula When none. Todos los atributos deben pertenecer a la tabla extendida de la tabla base encontrada. Los atributos que no cumplan esto, no serán “instanciables”, pues no se puede llegar a ellos.

En el ejemplo, se procede en este orden:

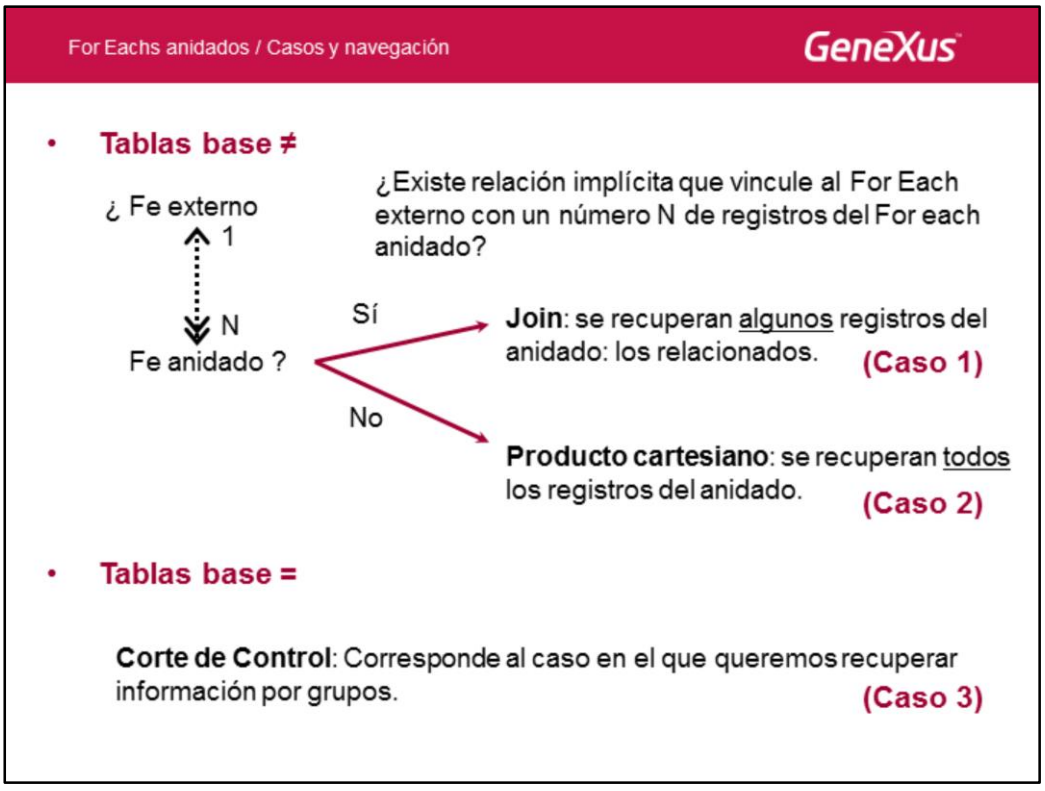
1) Se determina la **tabla base** del **for each externo**. Para eso se considera la transacción base indicada, o sea Country.City, y se verifica que los atributos presentes en el printblock (CountryName y CityName) pertenezcan a su tabla extendida. Si esto no es así, se produce una advertencia en el listado de navegación, que informará que algunos atributos no podrán instanciarse, pues no puede llegarse a ellos desde la tabla extendida de ese for each. En este caso CountryName y CityName pertenecen a la extendida de CountryCity, tabla base del for each.

2) Se determina la **tabla base** del **for each anidado**. Se considera la transacción base Attraction, y el atributo AttractionName presente en el printblock. No se toman en cuenta los atributos del for each exterior. Si no se hubiera escrito transacción base, entonces sí se consideraría algo relativo a los atributos del for each externo para determinar la tabla base del interno, pero no es el caso. Entonces se determina su tabla base como si fuera un for each independiente.

Por tanto, su tabla base será Attraction.

GeneXus

Casos y Navegación



De la determinación de las tablas base, surgen los tres casos de For eachs anidados que ya hemos visto 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**.
(En la página siguiente veremos otro caso, pero que responde a una mala programación).

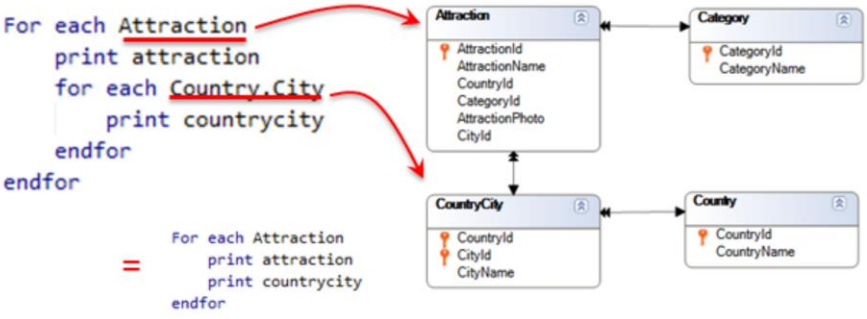
Cuando las **tablas base** son **la misma** se tratará de un caso conocido como Corte de control: es cuando necesitamos agrupar la información de una tabla, ejecutar ciertas instrucciones que tienen en cuenta la información común del grupo y luego recorrer cada miembro del mismo, y ejecutar otras instrucciones, 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**.

- **Tablas base ≠**

Fe externo
N
1
Fe anidado

Este es un caso raro, mal programado: cuando la tabla base del For each anidado es parte de la tabla extendida del For each externo.

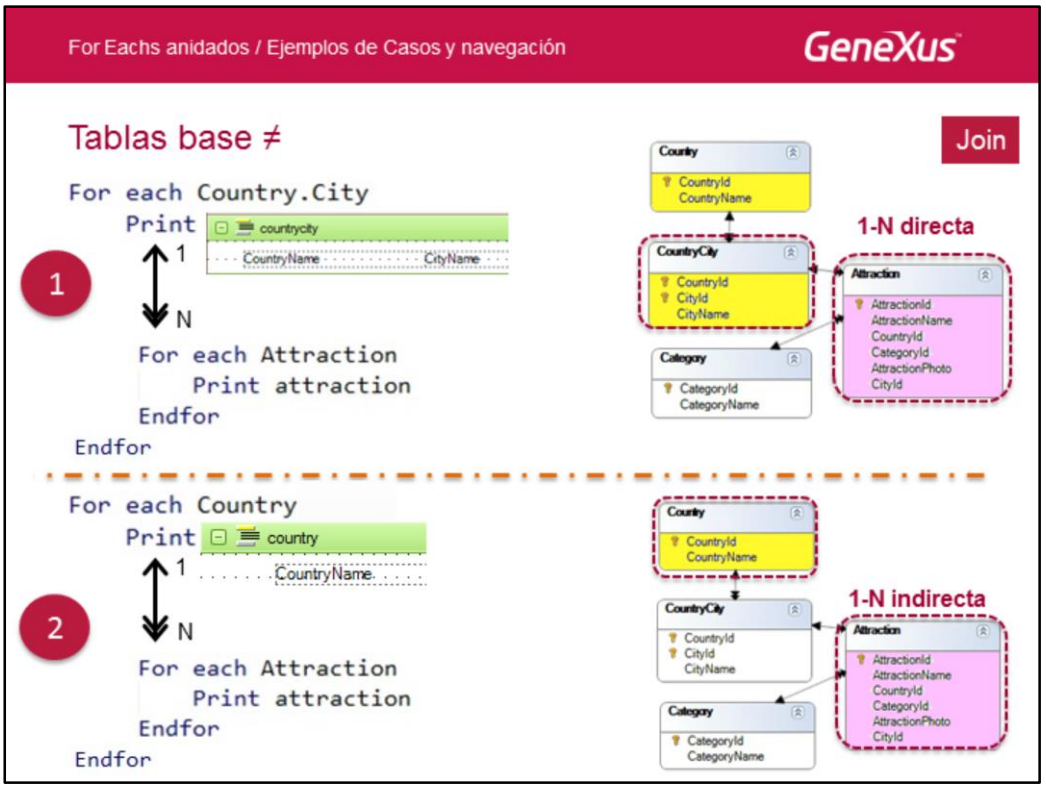
(Caso 4)



Este caso solamente surge cuando se utiliza transacción base para el for each anidado. De lo contrario, GeneXus tendrá que calcularla solo, y ante este caso elegirá para el for each anidado la misma tabla base que la del padre, implementando, entonces, un corte de control. No lo veremos en este curso.

Observemos que aquí el segundo for each sería innecesario, porque por cada atracción en la que se esté posicionado en un momento dado en el for each externo, únicamente se tiene un registro de CountryCity relacionado. Sería lo mismo, entonces, no haber escrito el segundo for each, y directamente enviar a imprimir el printblock countrycity, que contiene CountryName y CityName, ambos pertenecientes a la tabla extendida de Attraction.

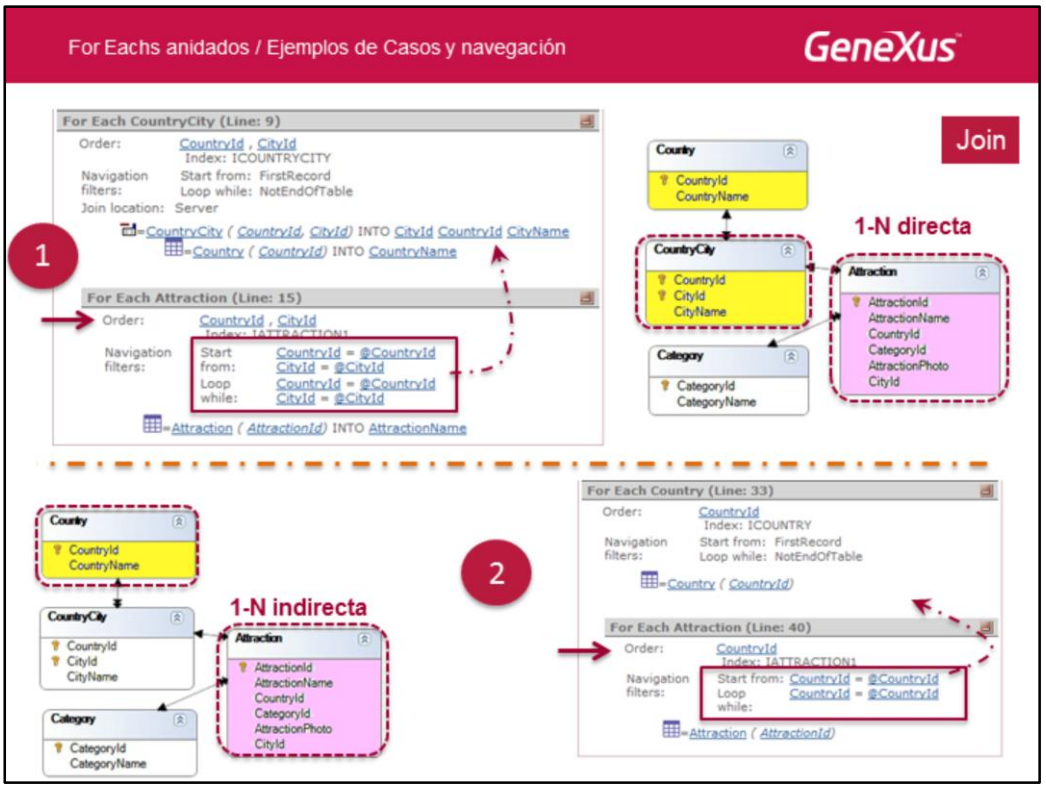
Ejemplos de cada caso



Aquí tenemos dos casos de relación 1 a N.

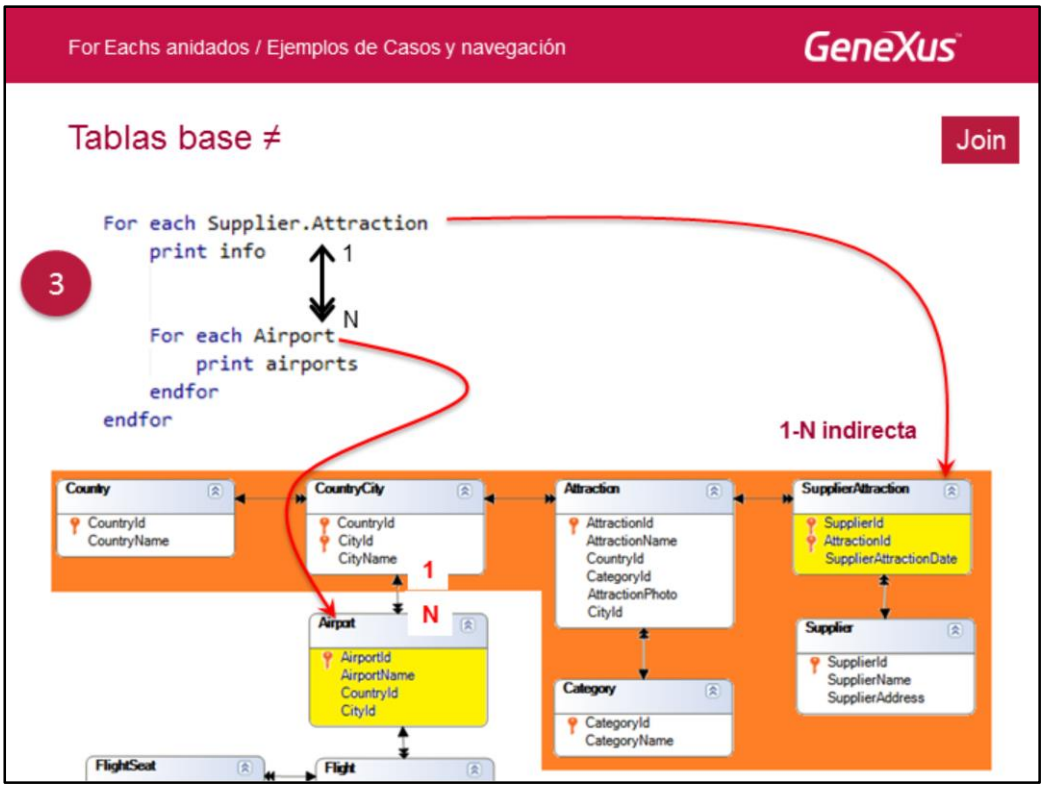
La primera es directa. Observemos que las tablas base del for each externo y anidado son CountryCity y Attraction, respectivamente, que están relacionadas por una relación 1 a N.

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).

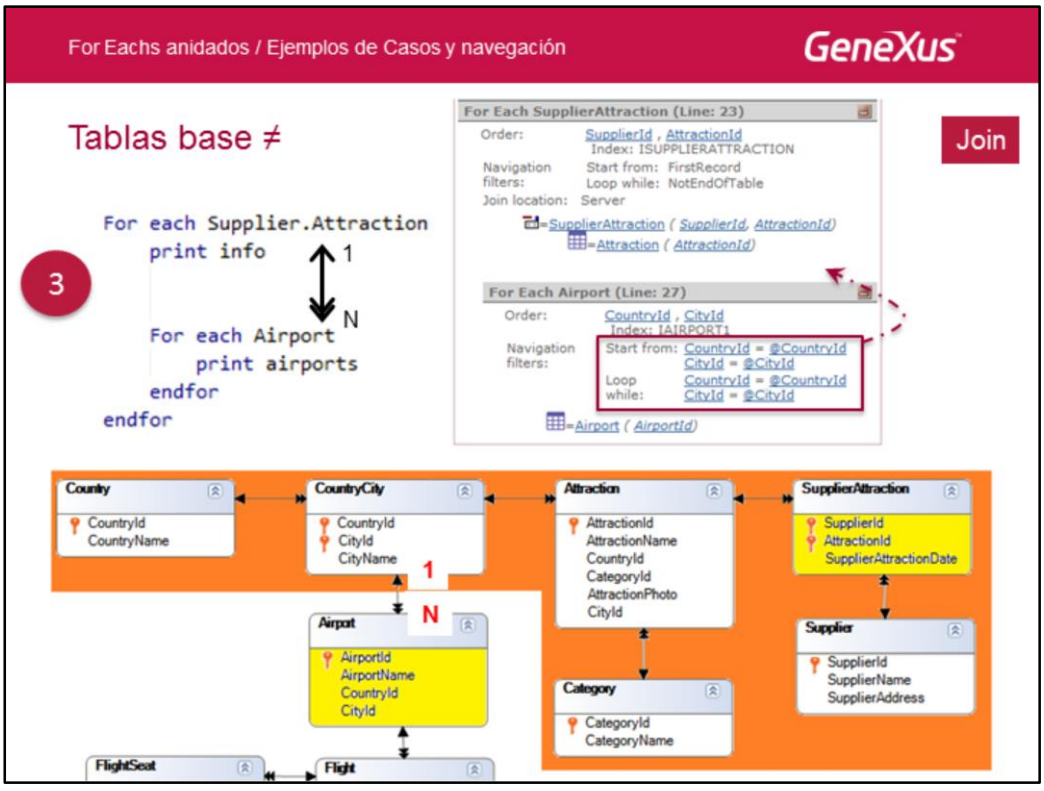


Los listados de navegación indican claramente el Join: para el for each anidado no se recorre toda la tabla. Obsérvese que en ambos casos, en lugar de ordenar la recorrida del for each anidado por la clave primaria de Attraction, que es AttractionId, lo hace por el atributo o conjunto de atributos relación,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

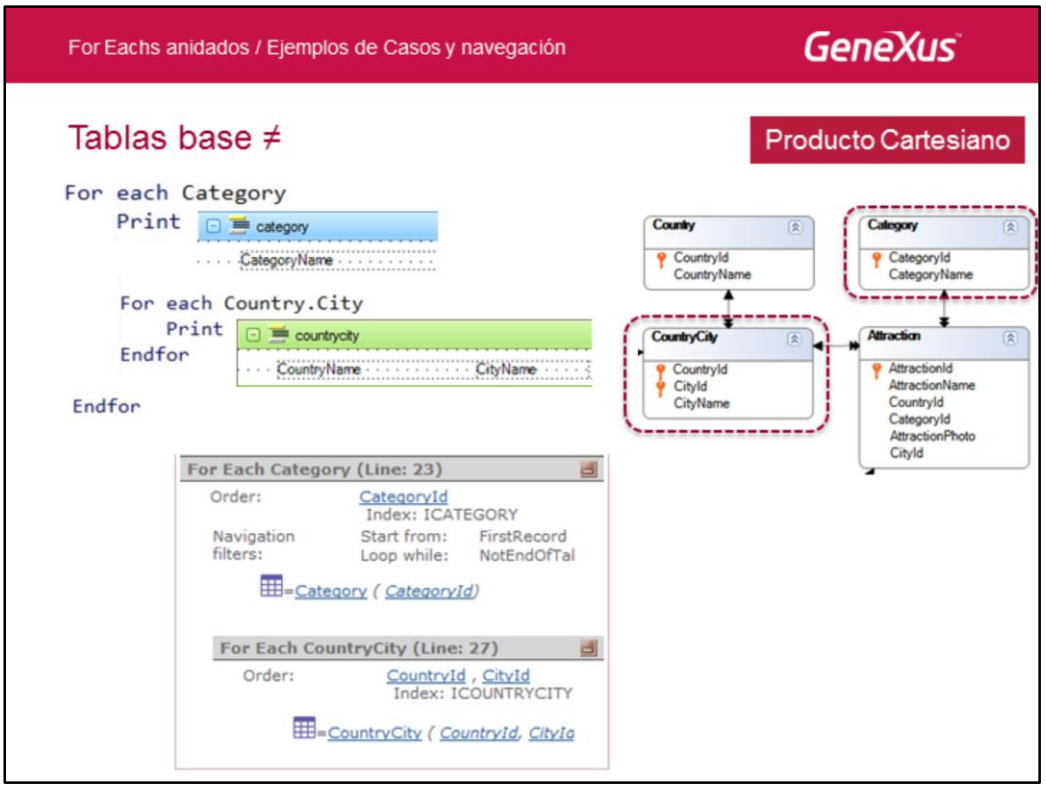
Por tanto, cuando GeneXus determina que realizará un Join, intenta optimizar su navegación.



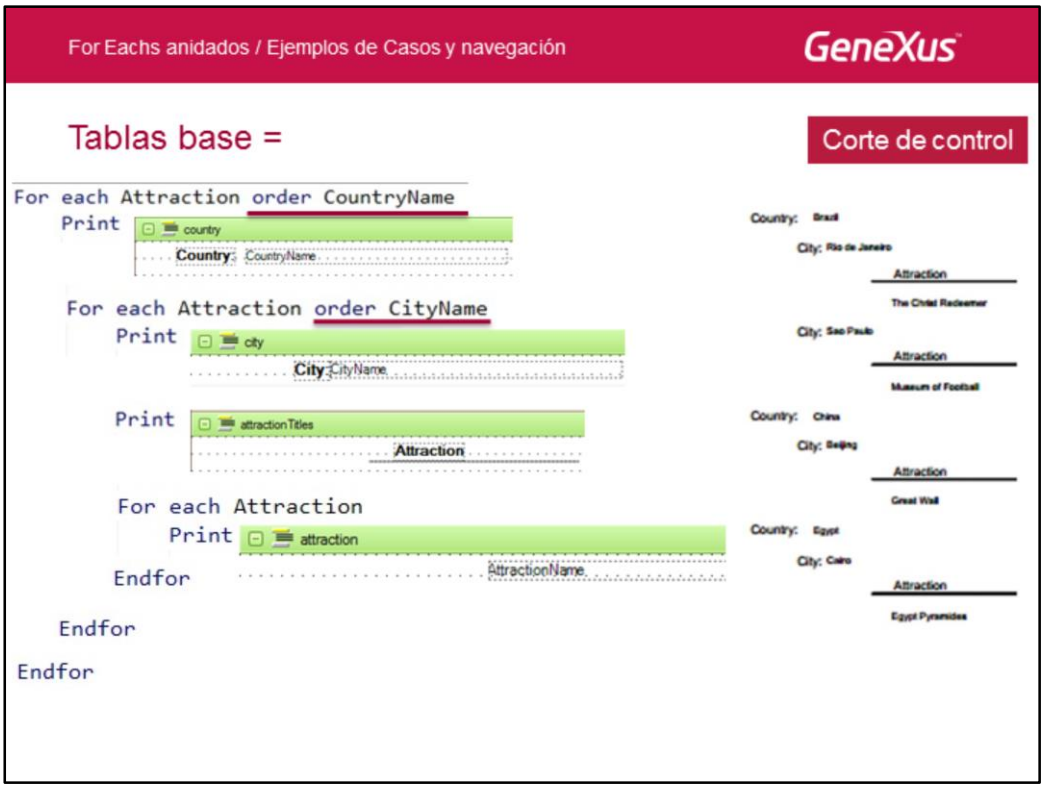
Aquí vemos un tercer ejemplo, donde la tabla base del for each externo es SupplierAttraction, y la tabla base del for each anidado en Airport. Observemos que existe una relación 1 a N indirecta. Es decir, por cada registro de SupplierAttraction, existirá uno solo de Attraction a partir del que obtenemos uno solo de CountryCity, que a su vez está relacionado con N registros de Airport (los N aeropuertos que se encuentran en ese país/ciudad, aunque en general, en la realidad, haya uno solo. En nuestro modelo puede haber varios).



Hilando fino, aquí GeneXus encuentra que hay atributos en común entre la tabla extendida del for each principal y la tabla base del anidado. ¿Cuáles? El par {CountryId, CityId}. Y por ellos realizará el join.



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).

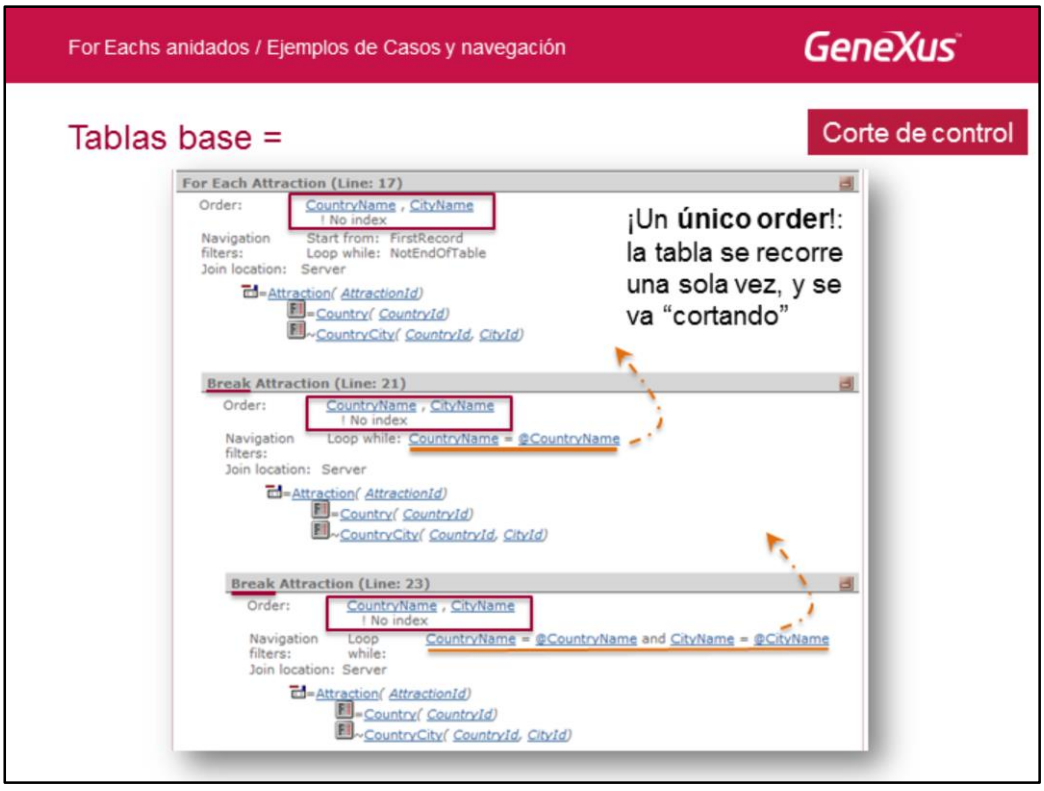


Aquí vemos un caso en que estamos queriendo listar cada país, y para él cada ciudad, y para ella cada atracción. La restricción: queremos hacerlo sólo para los países y ciudades para los que existen atracciones turísticas.

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:

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.



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

Si observamos el listado de navegación, 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 cuál 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. 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™

Videos	training.genexus.com
Documentación	wiki.genexus.com
Certificaciones	training.genexus.com/certificaciones