






Como listar informação relacionada

For eachs aninhados

GeneXus™

A listagem a seguir é solicitada:

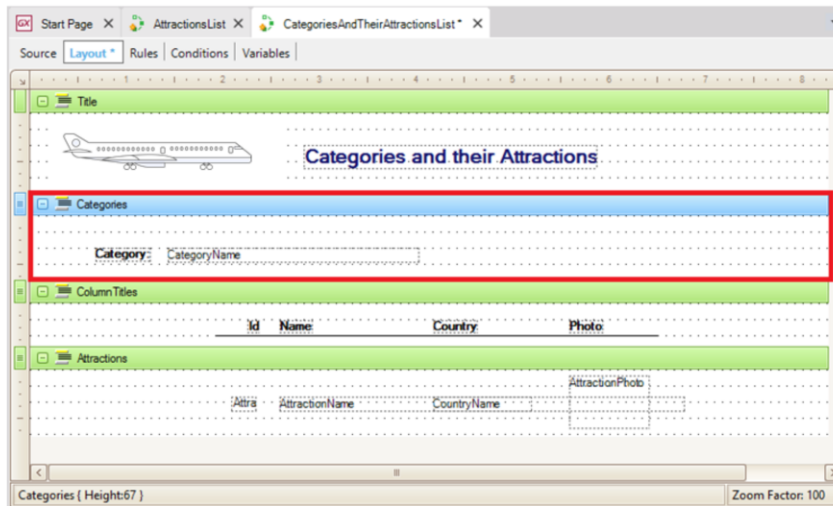
Título
 Categorias
 Títulos colunas
 Atrações

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 Redeemer	Brazil	
Category: Famous Landmark			
ID	Name	Country	Photo
2	The Great Wall	China	

Agora suponhamos que a agência de viagens nos solicita uma listagem que mostre todas as categorias de atrações turísticas e para cada categoria, todas as suas atrações.

Observemos que a grande diferença em relação à listagem de atrações que tínhamos implementado antes, é que agora queremos agrupá-las por categoria.

Layout



No Layout, adicionaremos um printblock aos que já tínhamos.

Chamaremos este novo printblock de Categories . E vamos inserir ali um Text Block... Category,
...e um atributo... CategoryName. Os outros print blocks permanecerão como estão.

Observemos que temos dois print blocks com conteúdo fixo: Title e ColumnTitles e dois print blocks com conteúdos variáveis, que terão que ser extraídos da base de dados: Categories e Attractions. Ambos contêm atributos. Categories tem CategoryName, da tabela CATEGORY e Attractions contém todos esses atributos, que já vimos que pertenciam à tabela estendida de ATTRACTION.

Agora vamos para Source.

Código fonte

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 temos que navegar as categorias e para cada uma delas **navegar várias** atrações (as que pertencem a essa categoria), esta lista é diferente da que já resolvemos anteriormente.

Antes:
For each Simples

Attraction
For each Attraction
print Attractions
endfor

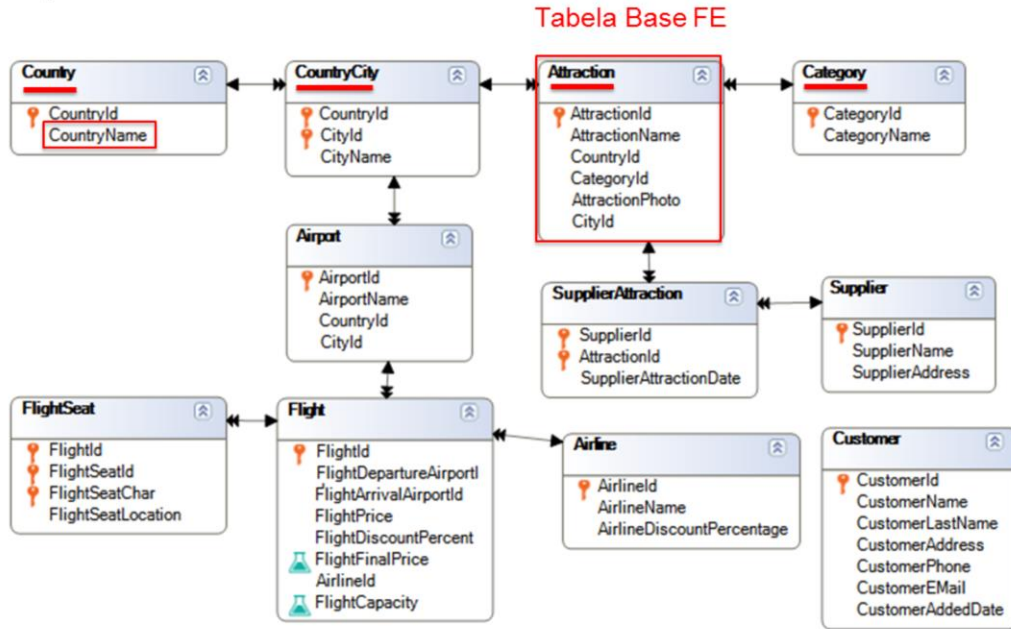


Tabela Base



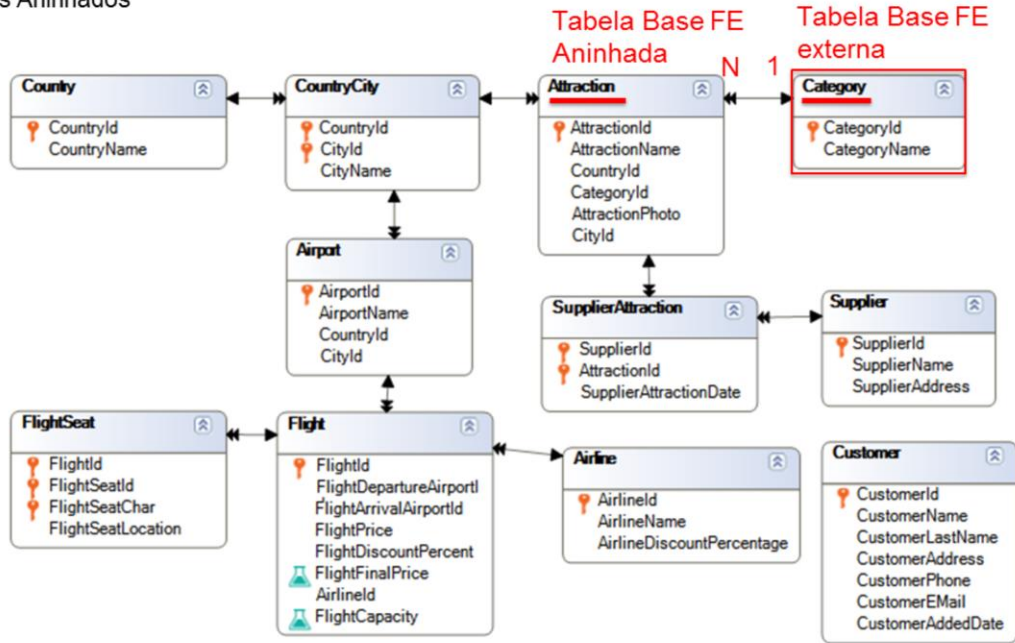
Na lista anterior navegamos as atrações e, como cada atração tem apenas um país, podíamos recuperar o nome do país de cada atração, por estar na tabela estendida da tabela base que estávamos navegando.

ANTES:
For each Simples



Sempre que a informação que queremos recuperar está disponível na tabela estendida da tabela de base que estamos navegando, **podemos fazer referência a ela diretamente no mesmo For each**. Foi o caso de CountryName.

AGORA:
For eachs Aninhados



Por outro lado, se estivermos navegando uma tabela –no nosso caso Category– e precisamos para cada registro acessado navegar **vários registros relacionados** que estão em outra tabela que não pertence à tabela estendida da tabela que estamos navegando ou percorrendo, como é nosso caso agora, com a tabela ATTRACTION, então precisaremos escrever **outro For each dentro primeiro, para percorrer o conjunto de registros relacionados**. Ou seja, teremos um for each **aninhado** dentro do outro.

Escrevendo o código

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

TABELA BASE: CATEGORY

TABELA BASE: ATTRACTION

São relacionados?

Aqui estamos posicionado em uma categoria (está instanciada)

CountryCity

Attraction

Category

Vamos então para o Source do nosso procedimento, e comecemos a escrever um primeiro For each para navegar e mostrar as categorias.

Ao lado do For each, escrevemos Category... Lembremos que aqui está a **transação base**, ou seja, o nome do nível da transação cuja informação queremos navegar.

Qual é a primeira coisa que queremos fazer com cada categoria acessada pelo for each? Imprimi-la. Dentro do For each, então, escrevemos: Print Categories

E como o printblock Categories só tem incluído o atributo CategoryName, e GeneXus deduziu que a **tabela base** do for each é CATEGORY e CategoryName está incluído na tabela estendida da tabela de base (porque neste exemplo está na própria tabela), então estará tudo em ordem e se poderá recuperar a informação corretamente. Caso contrário GeneXus dará um erro.

Depois que a categoria é impressa, queremos navegar no conjunto de **suas** atrações turísticas... por isso precisamos escrever o segundo For each, para percorrer as N atrações pertencentes à categoria que estamos navegando.

Mas antes de navegar o conjunto de atrações da categoria, teremos que mostrar os títulos que irão preceder essas atrações que vamos mostrar, então vamos escrever a instrução **Print ColumTitles**

E agora sim escrevemos o 2º For each, dentro do corpo do 1º ...

Em seguida, escrevemos Attraction, já que é o nome da transação cuja tabela associada queremos navegar agora, e dentro do For each, Print Attractions

Então, **Endfor** para fechar esta navegação e novamente **Endfor** para fechar a primeira.

Agora, **como GeneXus sabia quais atrações devia mostrar para cada categoria se nós não lhe indicamos explicitamente nada sobre isso?**

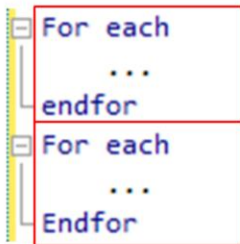
Observemos os for eachs. Sabemos que um for each percorre N registros de uma tabela e para cada um deles, executa uma série de instruções; que se encontram dentro do for each. Dentro desse “corpo” do primeiro for each, cada vez estaremos posicionados em uma e somente uma categoria. Dizemos que a categoria está **instanciada**, cada vez. É a categoria **tal**. Quando se terminam de executar as instruções do corpo, apenas lá se passa para a próxima categoria.

Então, antes de começar a executar o For each aninhado, GeneXus já sabe em qual categoria está posicionado nesse momento.

É por isso que escrevemos um For Each que navega pelas atrações, sem adicionar uma cláusula **where** para filtrar as atrações que atendam a condição de que sua categoria corresponda à categoria na qual estamos posicionados dentro do primeiro for each.

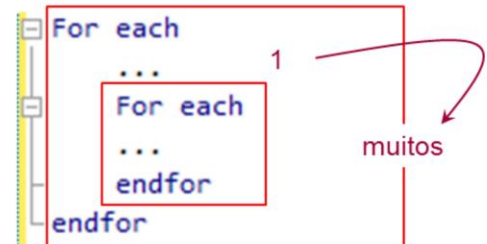
Filtro Implícito em For Eachs aninhados

For eachs Paralelos



Navegações
Independentes

For eachs Aninhados



Navegações
relacionadas?

Como fez GeneXus para estabelecer esse filtro sem que tivéssemos que escrevê-lo?

A resposta está na forma como escrevemos o For Each

Se dois comandos For each são escritos um após o outro, então eles **são independentes entre si**. Por outro lado, se escrevemos um For each **dentro** de outro, é porque, para cada registro da primeira navegação, queremos percorrer um conjunto de registros na segunda.

Quando escrevemos comandos For Each aninhados, GeneXus determina para cada For each, a tabela base que irá navegar... **e então busca se existe alguma relação entre essa informação.**

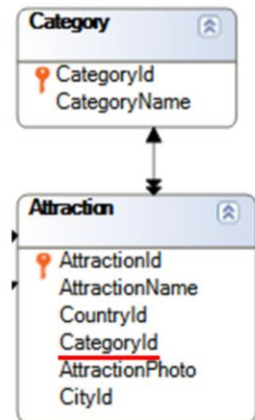
Filtro Implícito em For Eachs aninhados

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	



No nosso caso, a tabela base do For each externo é CATEGORY e a tabela base do For each interno é ATTRACTION e GeneXus tem o conhecimento de que existe um atributo em comum entre ambas as tabelas. Este atributo comum é **CategoryId**, que é chave primária em CATEGORY e chave estrangeira em ATTRACTION.

Portanto, o atributo CategoryId faz com que se relacionem as tabelas ATTRACTION e CATEGORY, como vemos aqui no diagrama, constituindo uma relação 1 a N. Isto é, para cada categoria, há muitas atrações relacionadas.

Filtro Implícito em For Eachs aninhados



Portanto, para cada categoria navegada no For Each externo, GeneXus executa o For Each que navega na tabela de atrações, **filtrando apenas aquelas atrações que cumprem que o valor de seu CategoryId coincida com o valor do CategoryId da categoria em que estamos posicionados.**

É **exatamente** como se no For each interno, tivéssemos escrito **Where CategoryId=CategoryId....** mas não precisamos colocá-lo, porque GeneXus o detecta e o aplica.

Lista de navegação

The screenshot displays the 'Navigation View' of a GeneXus project. The main window shows a 'Procedure CategoriesAndTheirAttractionsList Navigation Report'. The report is structured as follows:

- Procedure Information:**
 - Name: CategoriesAndTheirAttractionsList
 - Description: Categories And Their Attractions List
 - Output Devices: File
 - Main: Yes
 - Environment: Default (C#)
 - Spec. Version: 15_0_0-105189
 - Form Class: Graphic
 - 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
 - Table: Category (CategoryId) INTO CategoryId CategoryName
 - For Each Attraction (Line: 14):**
 - Order: CategoryId
 - Index: IATTRACTION2
 - Navigation Start from: CategoryId = @CategoryId
 - filters: Loop while: CategoryId = @CategoryId
 - Join location: Server
 - Table: Attraction (AttractionId) INTO CountryId AttractionPhoto.Uri AttractionPhoto AttractionName AttractionId
 - Table: Country (CountryId) INTO CountryName

A red arrow points from the '@CategoryId' filter in the 'For Each Attraction' section back to the 'CategoryId' field in the 'For Each Category' section, indicating the relationship between the two loops.

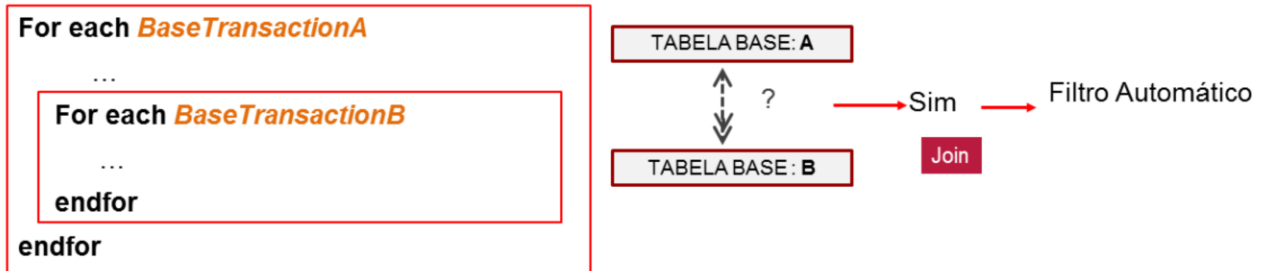
At the bottom of the window, the status bar shows: 0 Errors, 1 Warnings, 2 Success.

Se vamos à Lista de Navegação deste procedimento, vemos que nos informa sobre os dois for eachs, e vemos que a tabela base do externo é Category, e a do aninhado é Attraction. Além disso, podemos apreciar que as categorias são recuperadas ordenadas por seu identificador, CategoryId e que as atrações também são ordenadas por esse atributo, mas que nesta tabela é chave estrangeira, pois é o atributo que as relaciona, e é por isso que vemos nos filtros de navegação que somente irá recuperar as atrações desta categoria.

Vimos o quão fácil é obter informações e exibi-las em uma lista... mas os procedimentos podem fazer muito mais que isso. E veremos mais tarde.

Resumo

- Informação Relacionada



Como revisão, lembremos que, quando escrevemos comandos For Each aninhados, GeneXus determina para cada For each a tabela base que irá navegar e, em seguida, procura se há alguma relação entre essas tabelas base.

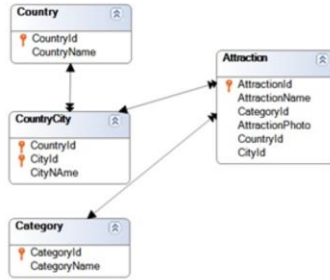
Se a resposta for sim, como vimos na listagem desta aula, aplicará então um filtro automático aos registros percorridos pelo for each aninhado. Neste caso de for eachs aninhados onde a informação é filtrada com base em um critério de relacionamento, chamamos-lhe **Join**.

Resumo

Join

```

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

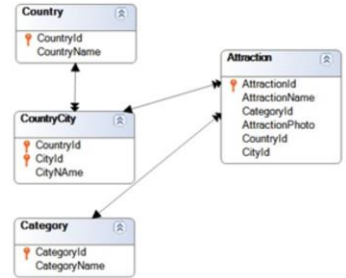


1-N directa

1-N indirecta

```

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



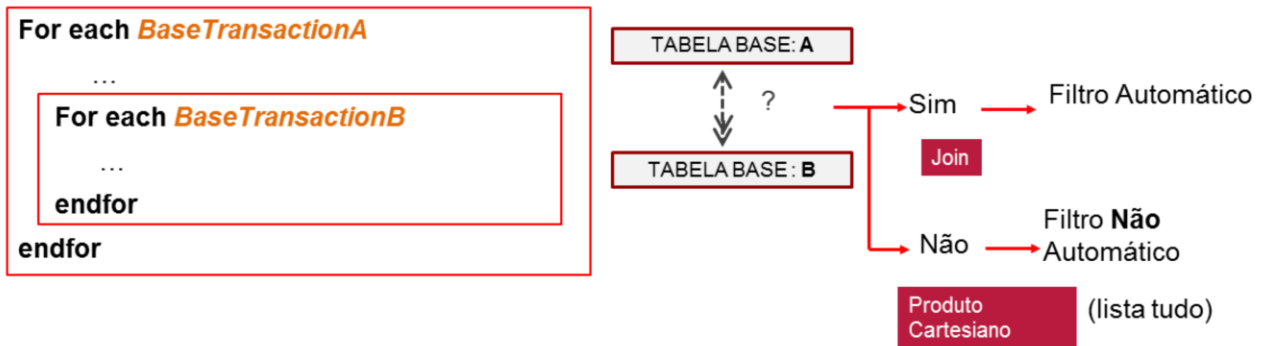
Aqui podemos distinguir dois casos de relacionamento 1-N entre duas tabelas.

O primeiro é direto. Observemos que as tabelas base do for each externo e aninhado são CountryCity e Attraction, respectivamente, que estão relacionadas por uma relação 1 para N.

O segundo é indireto. As tabelas base do for each externo e aninhado são Country e Attraction, que não têm uma relação direta 1 para N, mas sim indireta, através da tabela CountryCity. Em outras palavras: observemos que a tabela base do primeiro for each (Country), está incluída na tabela estendida da tabela base do for each aninhado (Attraction).

Resumo

- Informação Relacionada



Se, em vez disso, a resposta para a questão sobre se existe um relacionamento fosse Não, então nenhum filtro será aplicado. Serão impressos todos os registros do for each aninhado para cada cada registro do for each externo. A este tipo de for eachs aninhados em que nenhuma relação implícita é encontrada, chamamos de **Produto Cartesiano**. Claro, o desenvolvedor sempre pode especificar condições de filtro explícitas, escrevendo-as diretamente no for each com cláusulas where.

Nesses casos, supomos que as tabelas eram diferentes. Em seguida, veremos o que acontece quando as tabelas do for each externo e do aninhado são a mesma tabela.

GeneXus™

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