

# Procedures e Relatórios

Comando para acessar o banco de dados

**GeneXus**<sup>™</sup>

## Objeto Procedure

- ✓ Permite definir processos para acessar e navegar tabelas no banco de dados com vários objetivos ...



Em seguida veremos alguns objetivos possíveis

## Objeto Procedure

- ✓ Talvez seja necessário navegar nos registros de uma determinada tabela que cumpram determinadas condições e para esses registros atualizar um determinado atributo com um valor específico.

Tabela Attraction



AttractionId	AttractionName	Visites
1	Louvre Museum	8245
2	The Great Wall	10122
3	Eiffel Tower	11734

# Procedure object

- ✓ Navegar em determinada tabela e imprimir todos os seus dados em um PDF, ordenados por algum critério.

Tabela Attraction



AttractionId	AttractionName	CountryId	...
1	Louvre Museum	2	...
2	The Great Wall	3	...
3	Eiffel Tower	2	...

 ↓

**Attractions List**

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

## Objeto Procedure

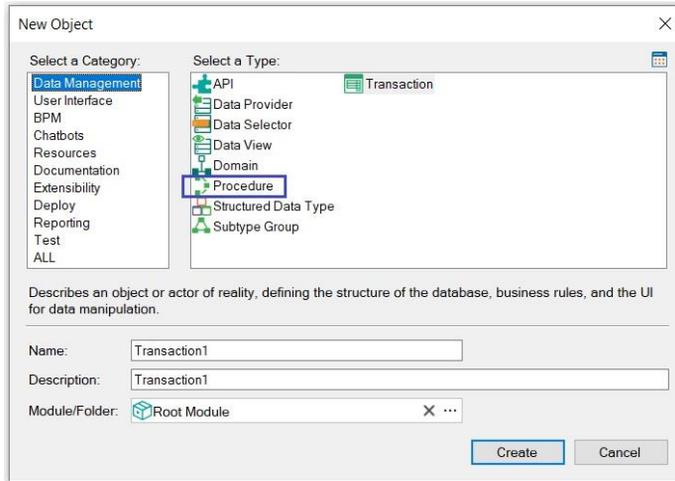
- ✓ Definir processos específicos que contenham pesquisas, cálculos, atualizações no banco de dados e, depois, imprimir essas informações.



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

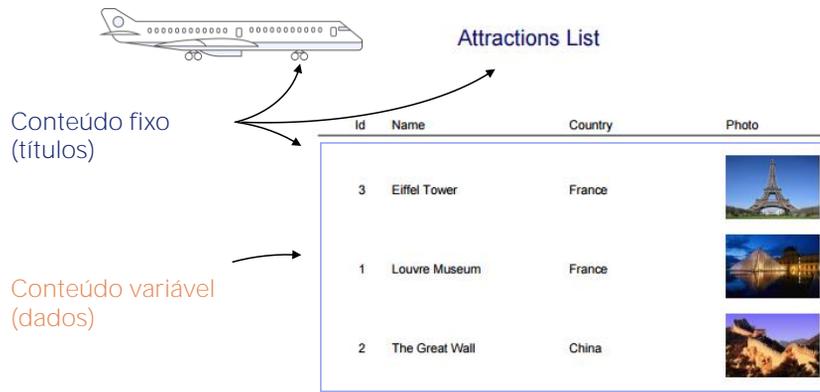
Resolvemos tudo isso criando Procedimentos em nossa KB (objeto Procedure do GeneXus).

# Objeto Procedure



Exemplo: listar dados em formato PDF

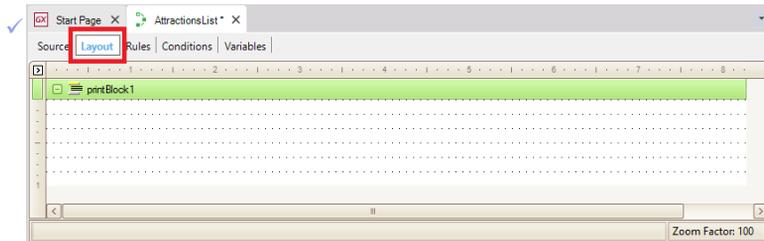
- ✓ Se quiser listar em um PDF todas as atrações turísticas da agência de viagens, em ordem alfabética.



Vamos começar definindo um procedimento para listar todas as atrações turísticas que a agência de viagens tem cadastradas, em ordem alfabética.

## Procedure: Source e Layout

- ✓ Source: para as instruções a serem executadas

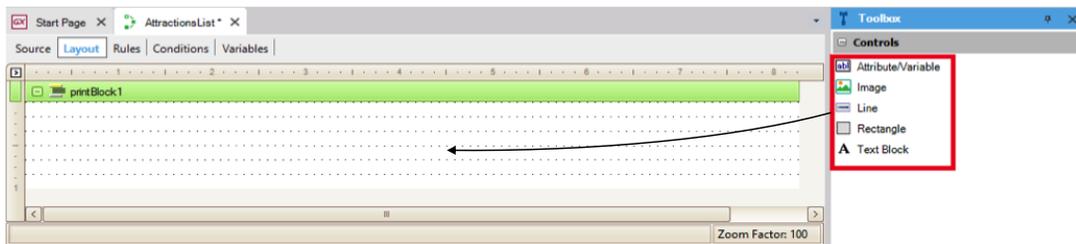


Depois que o objeto é criado, vemos que GeneXus cai automaticamente na seção chamada Source. Aqui é onde vamos digitar os comandos que farão com que o procedimento cumpra seu objetivo, que neste caso é imprimir uma lista de atrações turísticas.

Agora vamos dar uma olhada nesta outra seção chamada Layout. O layout é o local onde a saída é desenhada; Ou seja, onde definimos como queremos ver os nossos dados.

## Layout

- ✓ É composto de printblocks



- ✓ O primeiro já aparece por padrão. É chamado printBlock1.

É composto de printblocks e dentro desses printblocks é onde vamos incluir o que queremos mostrar.

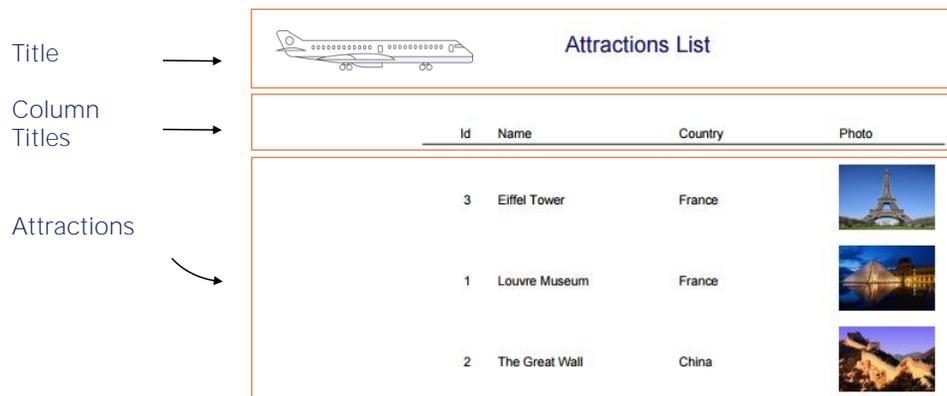
Podemos querer mostrar títulos, linhas, retângulos, imagens, bem como valores de atributos ou variáveis. Para fazer isso, vamos arrastá-los para o printblocks a partir da ToolBox.

Observe que um printblock é incluído automaticamente no layout.

Neste printblock, podemos exibir um título ou a data de hoje, por exemplo. Ainda podemos adicionar mais printblocks nesta seção, como veremos.

## Layout

- ✓ Printblocks para o nosso relatório:



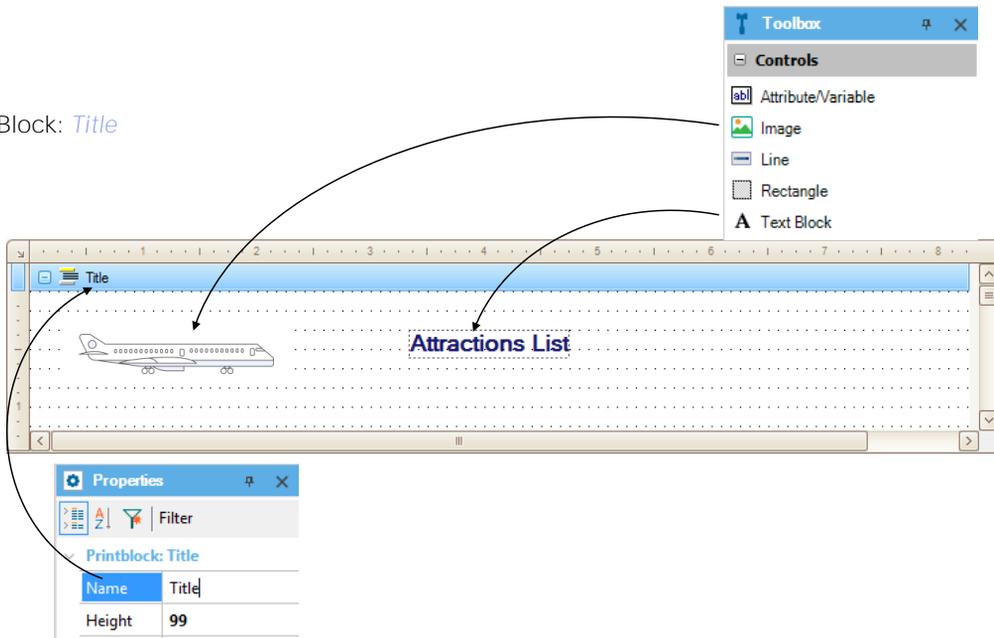
Podemos definir três printblocks:

- Um com o título do relatório e sua imagem, que pode ser chamado de Title
- Outro para mostrar os títulos das colunas com um linha embaixo deles, que pode ser chamado ColumnTitles
- e um terceiro printblock onde iremos exibir os detalhes da atração turística, que pode ser chamado Attractions.

Então, vamos começar a definir isso.

## Layout

- ✓ 1. PrintBlock: *Title*



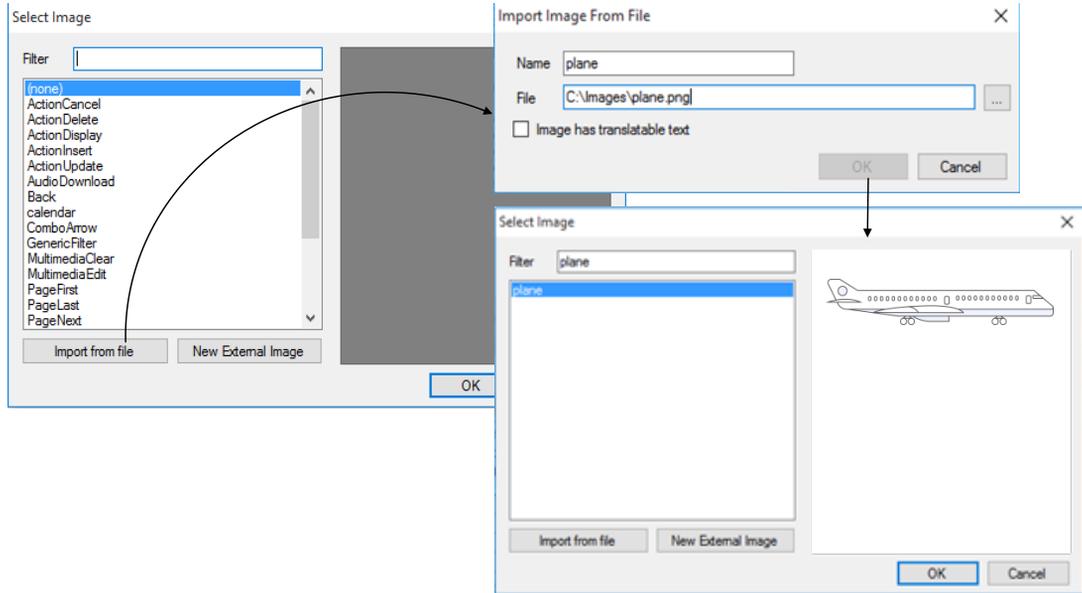
Para o título e a imagem, podemos usar o printblock que foi gerado quando criamos o objeto procedimento.

Vamos começar pelo título. Para fazer isso, arrastamos o controle TextBlock da Toolbox ... editamos suas propriedades ... e, na propriedade Text, escrevemos "AttractionsList". Também modificamos sua cor, para MidnightBlue, sua fonte ... Tamanho 14, Bold = Verdadeiro, e o colocamos onde queremos que ele apareça em relação às margens.

Vamos dar a este printblock um nome claro, que representa o que está mostrando. Então, selecionamos as propriedades do printblock e editamos sua propriedade Name, alterando seu valor para "Título".

Insira, agora, a imagem com o avião, à esquerda. Para fazer isso, arrastamos o controle Image da Toolbox até onde queremos colocá-lo .... Ao fazer isso ...

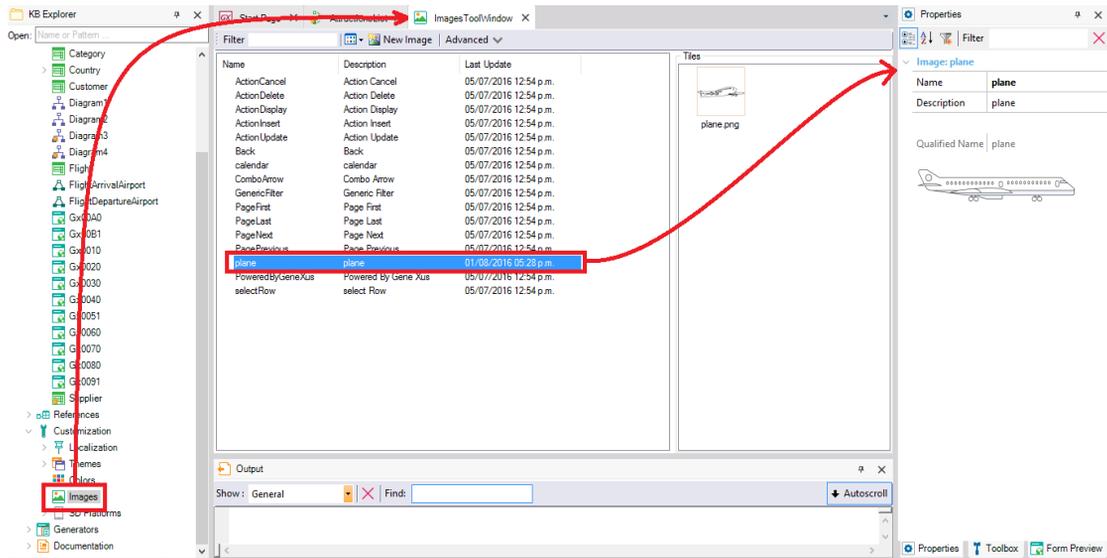
## Images na Knowledge Base



... é aberta uma janela que nos permite seleccionar algumas das imagens existentes na base de conhecimento ou incorporar uma nova, por exemplo, importá-la de um arquivo.

O botão "Import from file" nos permite explorar nosso sistema de arquivos e escolher a imagem, que será criada como um objeto GeneXus do tipo Image com o mesmo nome do arquivo de imagem, por padrão. A partir daí, podemos usar a imagem dentro do nosso KB livremente.

## Imagens na Knowledge Base



The screenshot shows the KB Explorer interface. On the left, the 'Images' folder is selected under 'Customization'. The main pane displays a list of images with columns for Name, Description, and Last Update. The 'plane' image is highlighted. The right-hand pane shows the properties for the selected image, including its name, description, and a small preview of the airplane image.

Name	Description	Last Update
Action/Cancel	Action Cancel	05/07/2016 12:54 p.m.
Action/Delete	Action Delete	05/07/2016 12:54 p.m.
Action/Display	Action Display	05/07/2016 12:54 p.m.
Action/Insert	Action Insert	05/07/2016 12:54 p.m.
Action/Update	Action Update	05/07/2016 12:54 p.m.
Back	Back	05/07/2016 12:54 p.m.
calendar	calendar	05/07/2016 12:54 p.m.
Combo/Arrow	Combo Arrow	05/07/2016 12:54 p.m.
Generic/Filter	Generic Filter	05/07/2016 12:54 p.m.
Page/First	Page First	05/07/2016 12:54 p.m.
Page/Last	Page Last	05/07/2016 12:54 p.m.
Page/Next	Page Next	05/07/2016 12:54 p.m.
Page/Previous	Page Previous	05/07/2016 12:54 p.m.
plane	plane	01/08/2016 05:28 p.m.
Poweredby/GeneXus	Powered by Gene Xus	05/07/2016 12:54 p.m.
select/Row	select Row	05/07/2016 12:54 p.m.

Properties for Image: plane

Name	plane
Description	plane
Qualified Name	plane

Observe que, a partir do KB Explorer / Customization / Images, podemos acessar todas as imagens do KB entre as quais está a do nosso avião.

## Layout

### ✓ 2. PrintBlock: *ColumnTitles*



Agora vamos criar outro printblock para incluir nele os títulos das colunas, com uma linha abaixo. Se pressionarmos o botão direito do mouse sobre determinado printblock e selecionarmos a opção **“Insert Printblock”**, um novo printblock será inserido embaixo dele.

A ordem dos printblocks no Layout não é importante, uma vez que não será necessariamente a ordem em que serão impressos. Definimos quando imprimir cada printblock no código que vamos escrever no Source do procedimento. Veremos isso em breve.

Agora, neste novo printblock, colocaremos o nome de **“ColumnTitles”**. Em seguida, neste novo printblock vamos inserir um TextBlock para cada texto que queremos mostrar como título de coluna.

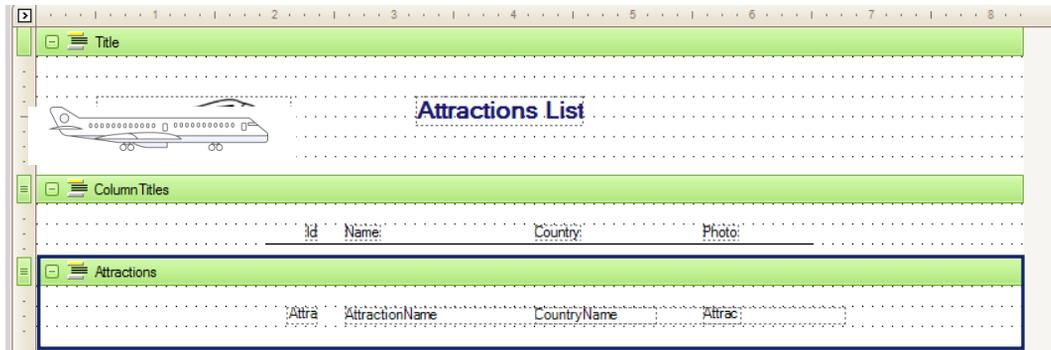
Então, arrastamos o text block da Toolbox e em sua propriedade Text escrevemos **“Id”**. Adicionamos outro Textblock, e em sua propriedade Text colocamos **“Name”**. Mais outro Textblock para mostrar o texto **“Country”**. E por último, um text block para o título **“Photo”**.

Colocamos os controles na posição que desejamos ... Nós podemos aliená-los selecionando todos eles e depois: Menu/Layout/Align/Bottom.

Finalmente, inserimos uma linha abaixo desses títulos de colunas. Então, voltamos para a ToolBox e arrastamos um controle **“Line”**. Nós arrastamos daqui ... dando o comprimento que queremos ...

## Layout

### ✓ 3. PrintBlock: *Attractions*



Agora, está faltando adicionar o terceiro printblock que havíamos falado, para mostrar os dados das atrações turísticas. Então, inserimos um novo printblock... e alteramos seu nome para Attractions.

Como os dados estão armazenados em atributos, vamos novamente até a Toolbox, selecionamos um controle do tipo “Attribute/Variable” e o arrastamos embaixo do título “Id”.

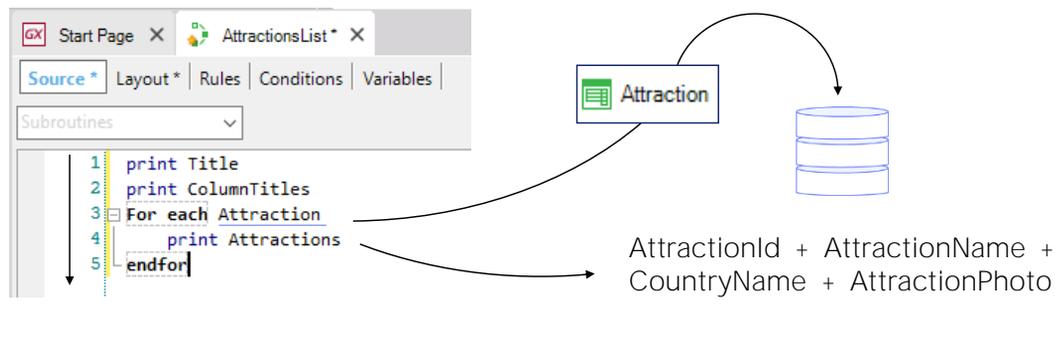
Na janela que se abre, escolhemos qual variável ou atributo queremos mostrar no controle. Podemos ver que, além de &today, em um procedimento existem outras variáveis do sistema.

Também podemos inserir atributos em um printblock através da opção **Insert / Attribute**

## Source

✓ Comandos:

- *Print* para imprimir um printblock e
- *For each* para percorrer uma tabela e sua estendida e fazer algo com cada registro encontrado



E está pronto o layout de como queremos que a informação seja exibida no relatório.

Agora, precisamos escrever o código necessário para obter as informações desejadas do banco de dados e para indicar em que ordem desejamos que os printblocks sejam impressos. Vamos até o Source ...

A primeira coisa que queremos imprimir é o título do relatório, então escrevemos "**print Title**". Como as instruções que escrevemos no Source serão executadas de cima para baixo, esta instrução será a primeira a ser executada. Estamos indicando com ela que deve ser impresso o conteúdo do printblock chamado Title, ou seja, o título do relatório.

O comando Print sempre deve preceder o nome de um printblock definido no Layout.

A próxima coisa que queremos é imprimir os títulos das colunas. Então, devemos escrever o comando para imprimir o printblock "ColumnTitles" ...

Com estas duas instruções, indicamos que a parte fixa do relatório seja impressa, ou seja, a parte que não irá variar de acordo com os dados: a que possui o título do relatório junto com a imagem do avião e também a que contém os títulos das colunas.

Mas agora precisamos imprimir a informação das atrações, que está armazenada no banco de dados. Para isso, devemos acessar a tabela física em que essa informação está armazenada, ou seja, a tabela associada à

transação Attraction.

O comando que nos permite acessar uma tabela física é o comando "For Each". A tabela física acessada é chamada de tabela base do for each.

Então, escreva o comando For Each ... e do seu lado: Attraction. Por que escrevemos Atração do lado do For Each? porque é o nome da transação cuja tabela física associada queremos navegar ...

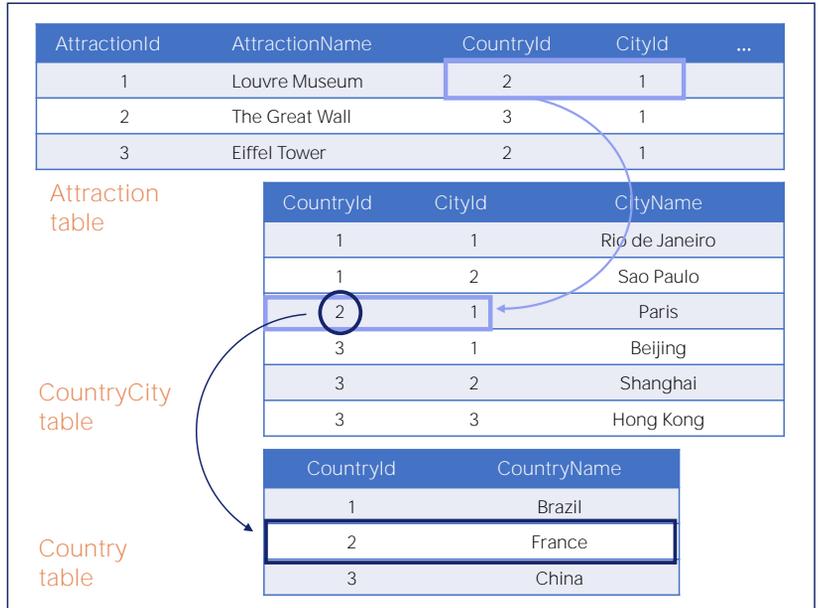
.. e agora ... uma vez que queremos imprimir o conteúdo dos atributos AttractionId, AttractionName, CountryName e AttractionPhoto de cada atração turística. Para isso escrevemos o comando para imprimir o printblock "Attractions" que os contém. Então, escrevemos Print Attractions.

Comando for each

For each



AttractionId + AttractionName +  
CountryName + AttractionPhoto

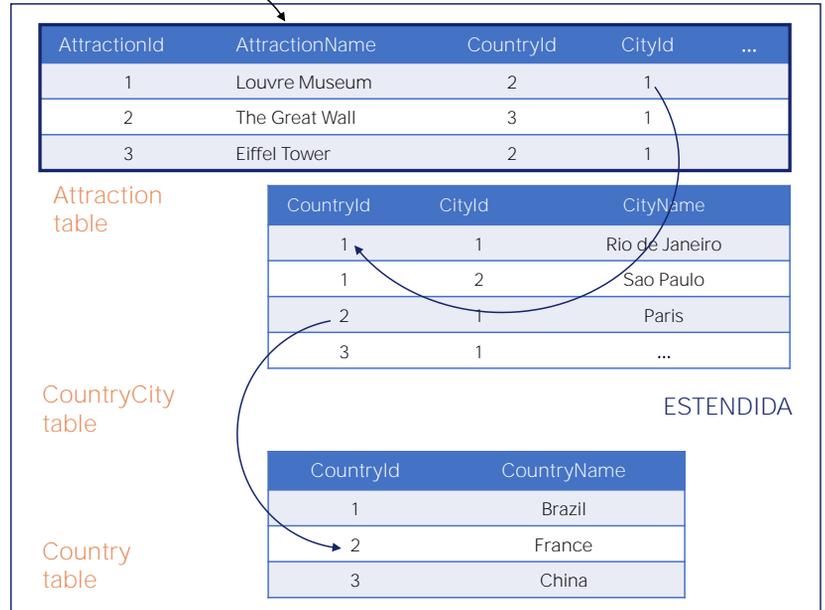
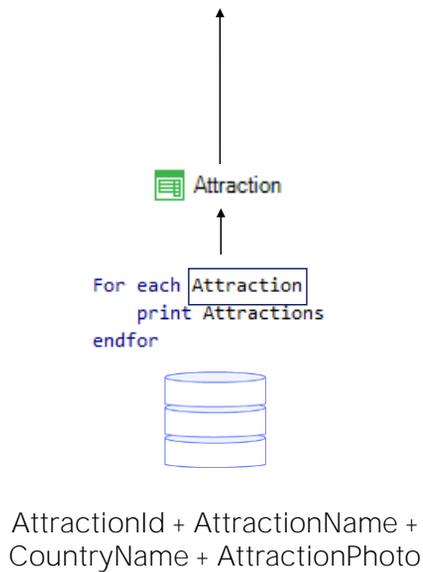


Desta forma, indicamos ao GeneXus que deve navegar na tabela física ATTRACTION, correspondente à transação Attraction.

E, como dentro do For each chamamos um printblock que contém atributos das tabelas ATTRACTION e COUNTRY, aplicando o conceito de tabela estendida, para cada atração navegada, acessaremos a tabela COUNTRYCITY, e a partir desta chegamos até PAÍS, para obter o nome do país onde a atração está localizada.

Transação Base

Tabela Base

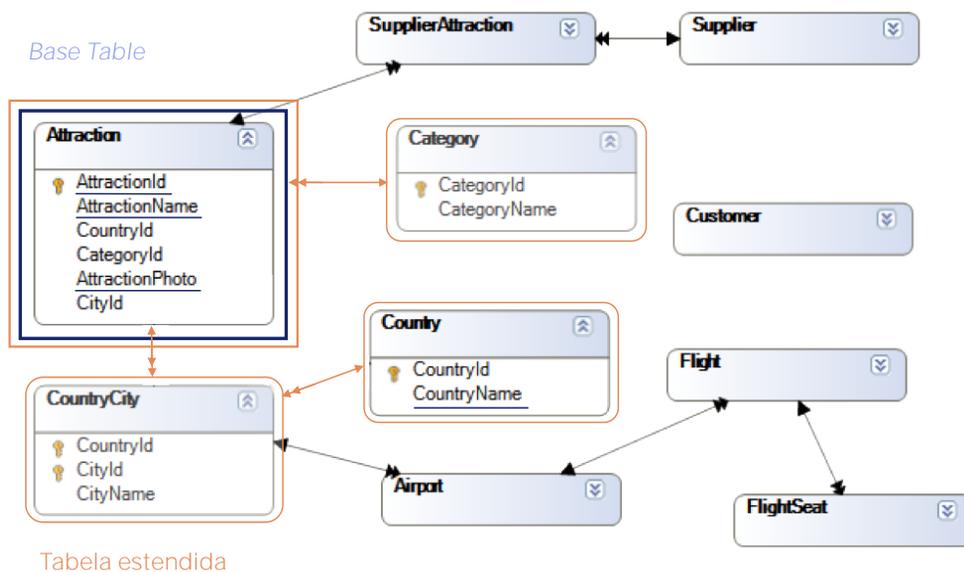


Vemos então que temos a **Transação Base**, indicada ao lado do For each. A partir da transação base, obtemos a **Tabela Base**, que será a tabela que o for each irá navegar, passando por seus registros um a um (neste caso, para imprimir um printblock).

Como no printblock não aparecem somente atributos da tabela Attraction, mas também de Country, para cada registro da tabela Attraction navegado durante a execução do for each, será acessado um registro de CountryCity e, a partir dele, a informação correspondente de Country, para recuperar o nome do país da atração.

Resumindo, os atributos utilizados no for each deverão pertencer à sua **tabela estendida**. Ou seja, à tabela estendida de sua tabela base.

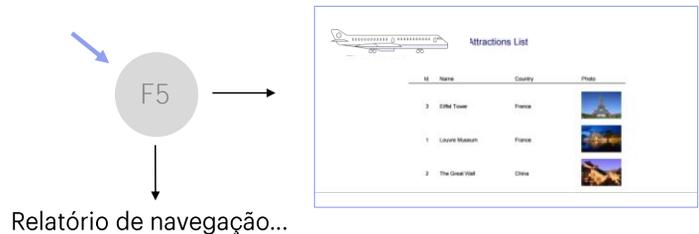
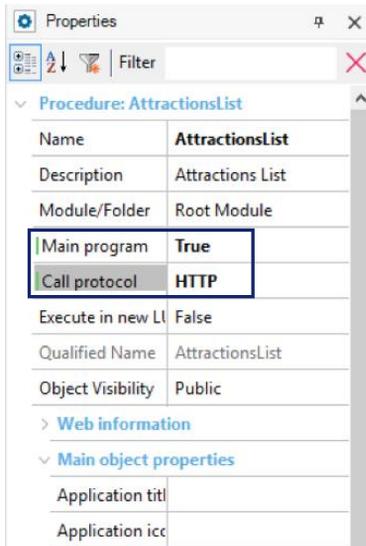
Transação base, Tabela Base e Tabela Estendida do For each



Aqui vemos um diagrama com os relacionamentos entre as tabelas de nossa base de conhecimento.

Dentro do for each temos os atributos AttractionId, AttractionName, AttractionPhoto e CountryName. Os três primeiros pertencem à tabela base do for each, enquanto que o último pertence a uma das tabelas estendidas.

## Como gerar relatórios em formato PDF



Vamos executar para ver o resultado.

Mas, primeiro temos que definir algumas propriedades necessárias para que o relatório possa ser impresso em formato PDF. Vamos até as propriedades do relatório e na propriedade “Main program” selecionamos True.

Depois, na propriedade “Call protocol” selecionamos “HTTP”.

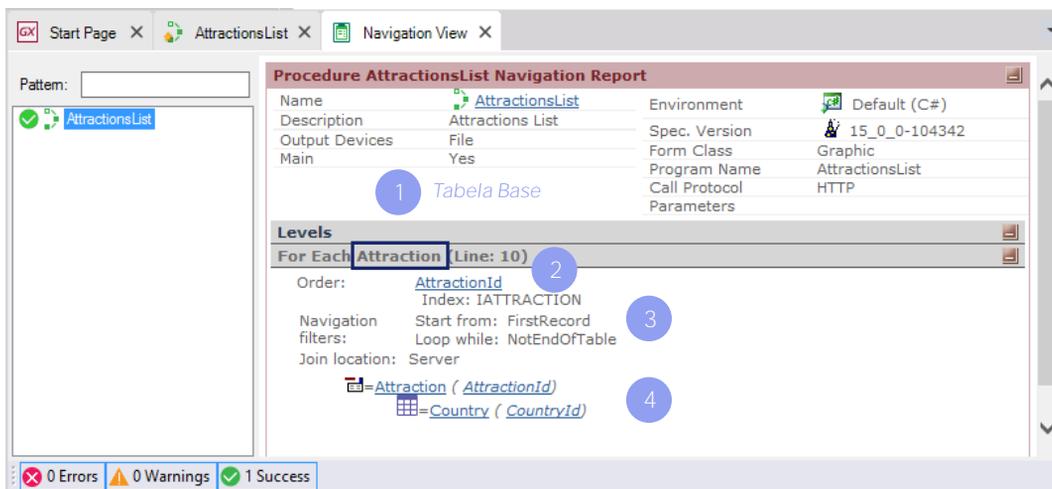
E, finalmente, temos que inserir a regra Output\_file na seção de regras ... como podemos ver, um objeto desse tipo também permite definir algumas regras - embora bem menos do que em uma transação. Então, selecionamos Insert/Rule... E completamos colocando o nome do arquivo como “AttractionsList.PDF” e depois o formato que vamos utilizar: “PDF”.

Salvamos... e agora já podemos executá-lo.

E veremos que a lista é criada! ... com o formato que definimos ... listamos todas as atrações turísticas que cadastramos, cada uma com o nome do país ao qual pertence e a foto.

Além disso, o GeneXus abre uma janela de “Navigation View” com um relatório...

## Relatório de navegação



A tabela física que o For each irá navegar, bem como outras decisões que GeneXus toma, são mostradas no Relatório de navegação do procedimento.

Este relatório é criado automaticamente quando o procedimento a ser executado é gerado. No nosso caso, foi depois do F5.

Nela, GeneXus nos diz como está acessando as informações no banco de dados.

1. Podemos ver que ao lado de onde ele diz "For Each", está escrito Attraction, para indicar que esta é a tabela base do For Each.

Vale lembrar que o For each percorre uma tabela física e assim, o nome Attraction que aparece no relatório de navegação é da tabela física ATTRACTION, não da transação base que escrevemos no procedimento. GeneXus deduz esta tabela porque é a tabela associada a essa transação base que indicamos.

2. Ele também indica que o atributo AttractionId (que é a chave primária da tabela Attraction) foi usado para ordenar a lista de atrações ...

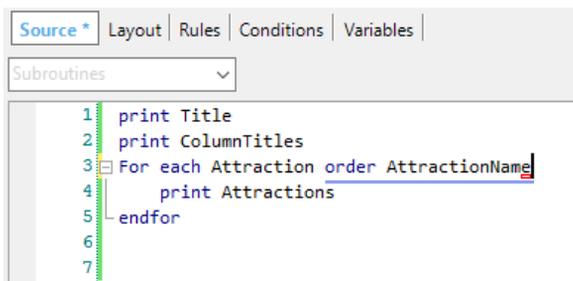
3. Mostra também que percorreu todos os registros da tabela: começou do primeiro registro e navegou até chegar ao fim da tabela. Mostrou todas as atrações....

4. e, finalmente ... indica que a tabela navegada foi Attraction e que teve que acessar o País para recuperar informações, pois na nossa lista mostramos o nome do país.

## Como alterar a ordem dos dados

- ✓ Foi pedido:

gerar um arquivo PDF com todas as atrações turísticas da agência de viagens, ordenadas alfabeticamente.



```
Source * | Layout | Rules | Conditions | Variables |
Subroutines
1 print Title
2 print ColumnTitles
3 For each Attraction order AttractionName
4   print Attractions
5 endfor
6
7
```

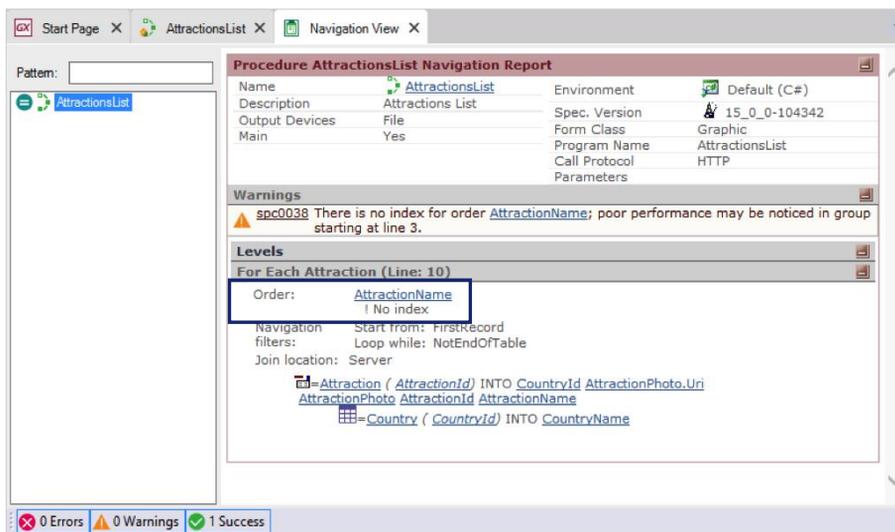
- ✓ É possível ordenar por qualquer atributo que pertença à tabela estendida da tabela percorrida pelo for each.

```
For each Attraction order CountryName
  print Attractions
endfor
```

Algo que ficou pendente foi listar as atrações em ordem alfabética, de acordo com o nome da atração.

Podemos resolver isso simplesmente adicionando a cláusula "order AttractionName" depois de "For each Attraction"

## Relatório de navegação

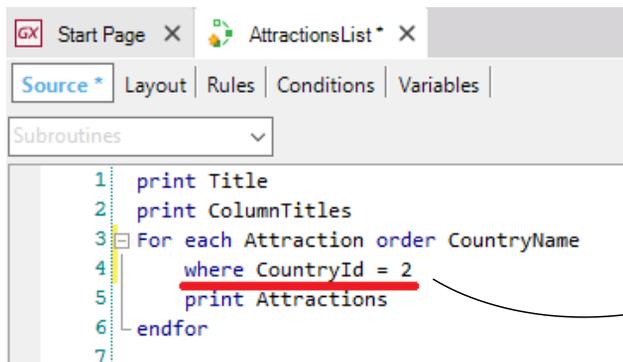


Por enquanto, desconsiderem o aviso que o relatório está mostrando. Mas, observe que o relatório de navegação nos informa qual atributo será responsável pela ordenação.

Assim como adicionamos a cláusula opcional “order” no For each, a sintaxe do For each permite adicionar várias outras cláusulas e definições opcionais, como veremos.

## Como definir filtros

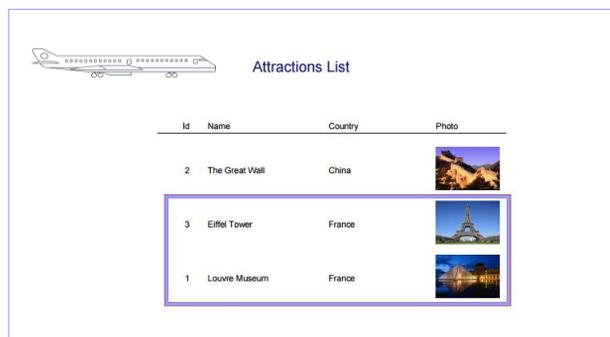
- ✓ Solicitação: listar todas as atrações turísticas da França.



```

1 print Title
2 print ColumnTitles
3 For each Attraction order CountryName
4   where CountryId = 2
5   print Attractions
6 endfor
7

```



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

where CountryName = "France"

Listado de navegación...

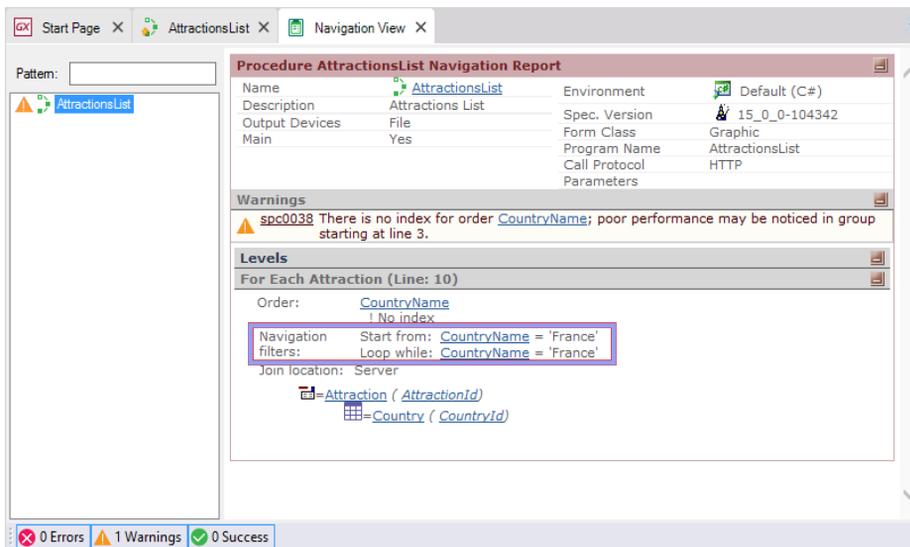
O que aconteceria, por exemplo se, na agência de viagens, nos pedirem para imprimir somente as atrações turísticas da França?

Basta acrescentamos uma cláusula **Where** no comando For Each, para que filtre e mostre somente os dados que cumpram com a condição desejada.

Então, nos posicionamos na linha abaixo do For Each e escrevemos Where ... CountryId = 2, pois sabemos que o ID da França é 2.

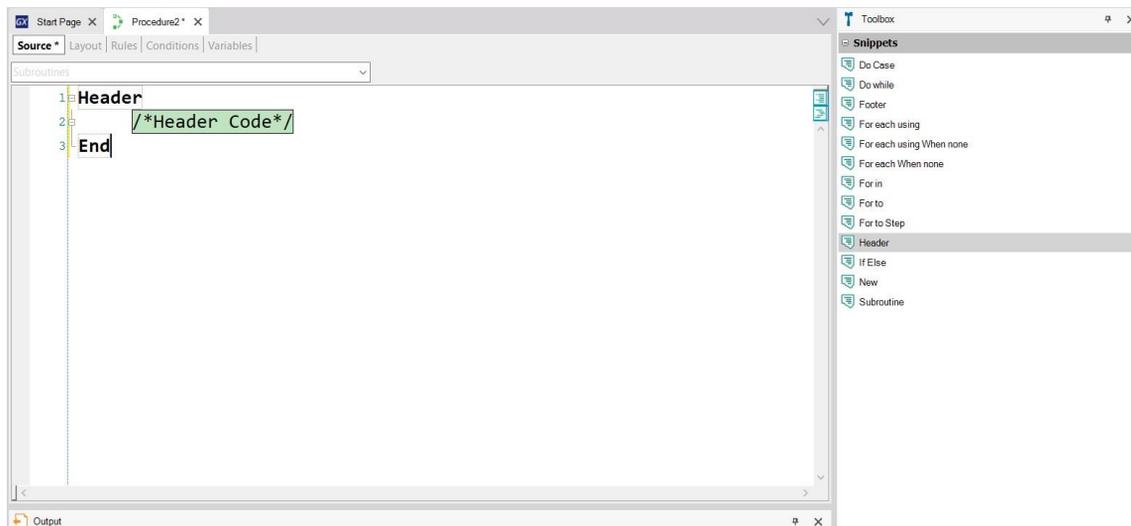
Em vez de filtrar pelo identificador do país, poderíamos ter escrito **Where CountryName='France'**.

## Como definir filtros: relatório de navegação



Vemos que não está mais sendo percorrida toda a tabela Attractions. Como estamos ordenando por CountryName, para separarmos os países com nome 'France' basta percorrer somente um pedacinho da tabela e não toda ela. É parecido com procurar a palavra 'France' em um dicionário. Não procuramos em todo o dicionário. Pulamos diretamente para o "F".

## Snippets



Antes de revisar tudo o que vimos, lembremos que o IDE de GeneXus é contextual, e isso significa que as propriedades que são visualizadas a partir da Toolbox dependem do objeto e do lugar onde estamos posicionados.

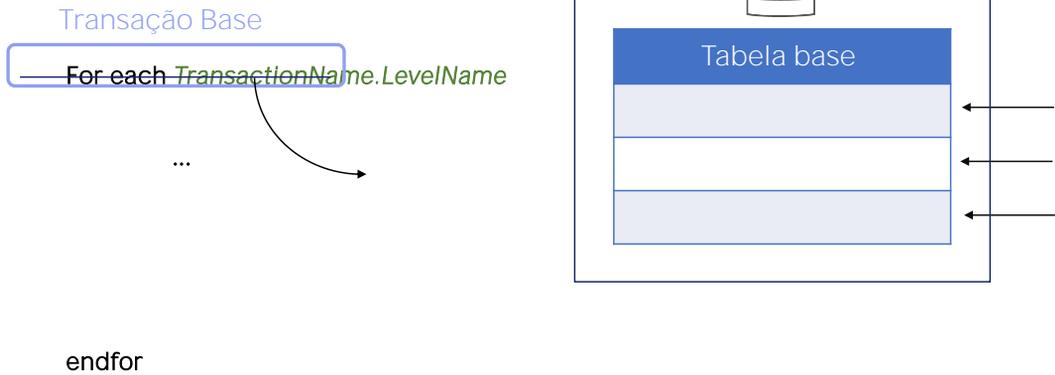
Se nos posicionamos no source de um procedimento, a Toolbox mostra os chamados Snippets, pedaços de código que oferecem a sintaxe geral das diferentes cláusulas e estruturas que podem ser utilizadas no source de um procedimento.

Entre eles existem comandos para controlar o desenho do pdf, como por exemplo, o Header que permite definir um cabeçalho que se repetirá em cada nova página da lista, ao contrário do título que colocamos, que só aparece na primeira página.

Sugerimos que você pesquise na wiki de GeneXus todas as opções para o desenho de uma lista.

## Conceito

- ✓ O comando For each é utilizado para percorrer cada registro de uma tabela e fazer alguma coisa com a informação relacionada.



O comando For each é utilizado para percorrer cada registro de uma tabela e fazer alguma coisa com a informação relacionada. Para isso, indicamos o nome da transação, ou mais precisamente, o nome do nível da transação cuja tabela associada queremos percorrer.

Chamamos esta indicação de nível de **transação base** do for each

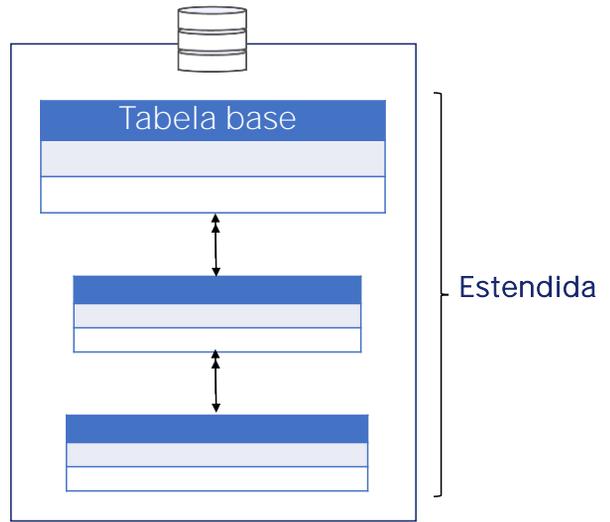
E, a partir desse nível, GeneXus irá inferir a tabela a ser percorrida, à qual chamamos **tabela base** do for each.

# Conceito

**For each** *TransactionName.LevelName*

Os atributos utilizados aqui devem pertencer à *tabela estendida* da tabela base a ser percorrida.

**endfor**



O conjunto de atributos que estão entre o For Each e o Endfor deve pertencer à tabela estendida da tabela base a ser percorrida.

## Conceptualizing

## ✓ Transação Base:

- 1er nível trn = **TransactionName**

**For each Flight**

**Where** FlightDepartureAirportId = 1

...

**Endfor**

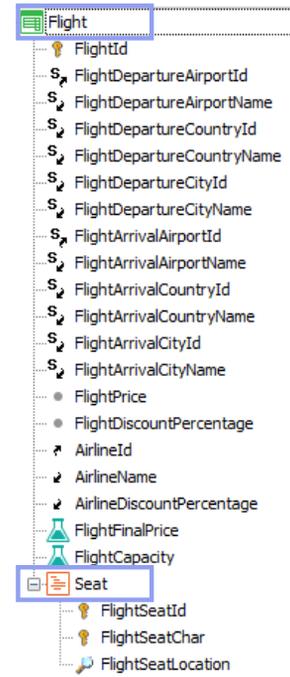
**For each Flight.Seat**

**Where** FlightId = 15

...

**Endfor**

- Nível aninhado trn = **TransactionName.LevelName**



No primeiro exemplo, queremos navegar a tabela de vôos que partem do aeroporto

1. Para um primeiro nível, o nome da transação coincide com o do nível.

No segundo exemplo, queremos navegar nos assentos do vôo 15.

## Sintaxe do For each

```
For each BaseTransaction
  order att1, att2, ..., attn
  where condition1
  where condition2
  ...
  where conditionn
    MainCode
endfor
```

Aqui resumimos o que foi visto até agora sobre o comando For Each.  
Vamos expandir essa sintaxe à medida que vermos mais tópicos.

## Sintaxe do For each: order

```
For each BaseTransaction
```

```
  order att1, att2, ..., attn
```

```
  where condition1
```

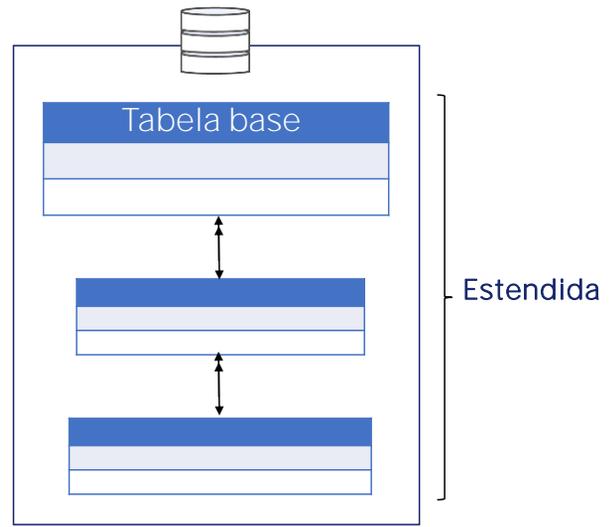
```
  where condition2
```

```
  ...
```

```
  where conditionn
```

```
    MainCode
```

```
endfor
```



Exemplo: [ORDEN COMPUESTO](#)

Através da cláusula **Order**, é possível indicar os critérios para ordenar a informação devolvida pelo For Each. A ordem pode ser definida com os atributos da tabela base do For each ou de sua estendida.

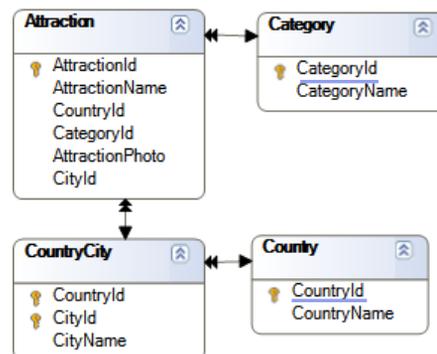
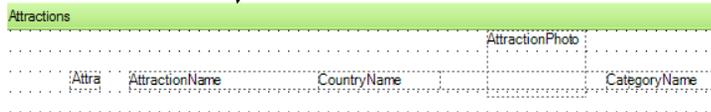
Como podemos ver, ele pode ser ordenado por um único atributo, ou por vários.

## Exemplo ordenação composta

```

print Title
print ColumnTitles
For each Attraction order CountryName, CategoryName
  print Attractions
endfor

```



Attractions List

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

Por exemplo, se também mostrarmos no relatório a categoria da atração turística e quisermos ordená-la pelo nome de país e, dentro das atrações que são do mesmo país, ordenar por nome de categoria ... nós escreveríamos ambos os atributos em sequência: primeiro CountryName e depois CategoryName ...

Aqui, tanto CountryName como CategoryName não estão presentes na tabela base, Attraction, mas nas tabelas estendidas.

Sintaxe do For each: where

**For each** *BaseTransaction*

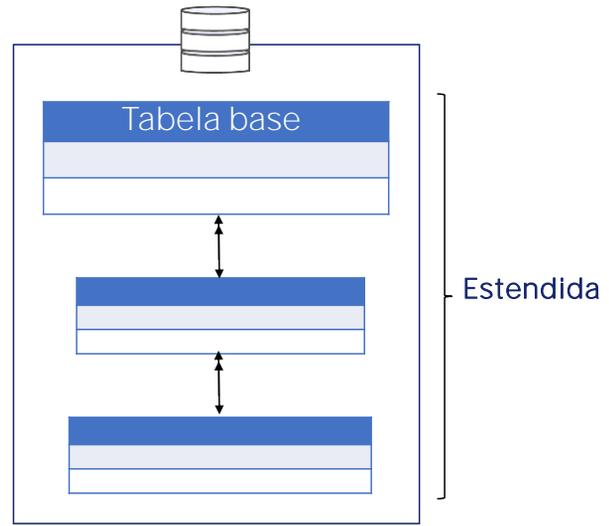
**order** *att<sub>1</sub>, att<sub>2</sub>, ... , att<sub>n</sub>*

**where** *condition<sub>1</sub>*

*condition<sub>a</sub> and condition<sub>b</sub>*  
*condition<sub>a</sub> or condition<sub>b</sub>*

*MainCode*

**endfor**



Para filtrar as informações devolvidas pelo For each usamos a cláusula **Where**, na qual especificamos a condição que os registros devem cumprir para serem selecionados.

A condição pode ser complexa, incluindo várias condições ligadas por AND ou OR, por exemplo:

- Conditiona **AND** Conditionb: significa que ambos devem ser cumpridos ao mesmo tempo
- Conditiona **OR** Conditionb: significa que, se uma das duas for cumprida, é suficiente para que o registro seja selecionado pelo filtro:

Sintaxe do For each: where

**For each** *BaseTransaction*

**order** *att<sub>1</sub>, att<sub>2</sub>, ... , att<sub>n</sub>*

**where** *condition<sub>1</sub>*      **and**

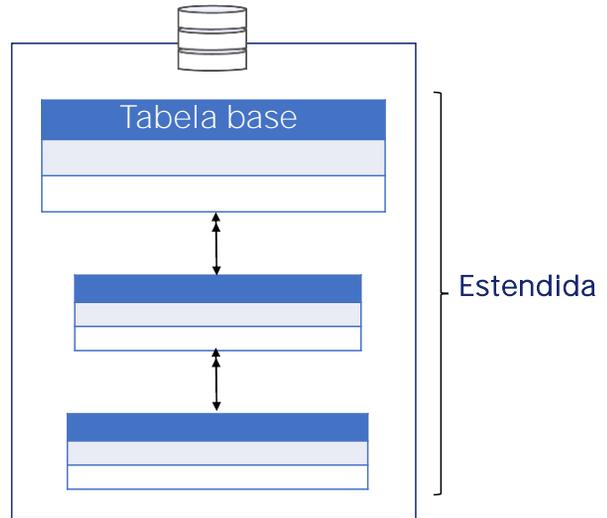
**where** *condition<sub>2</sub>*      **and**

...

**where** *condition<sub>n</sub>*      **and**

*MainCode*

**endfor**



Também podemos colocar várias cláusulas **Where**, que é o mesmo que escrever uma única cláusula, com as condições ligadas por **AND**:

## Sintaxe do For each: Main Code

**For each** *BaseTransaction*

**order** *att<sub>1</sub>, att<sub>2</sub>, ... , att<sub>n</sub>*

**where** *condition<sub>1</sub>*

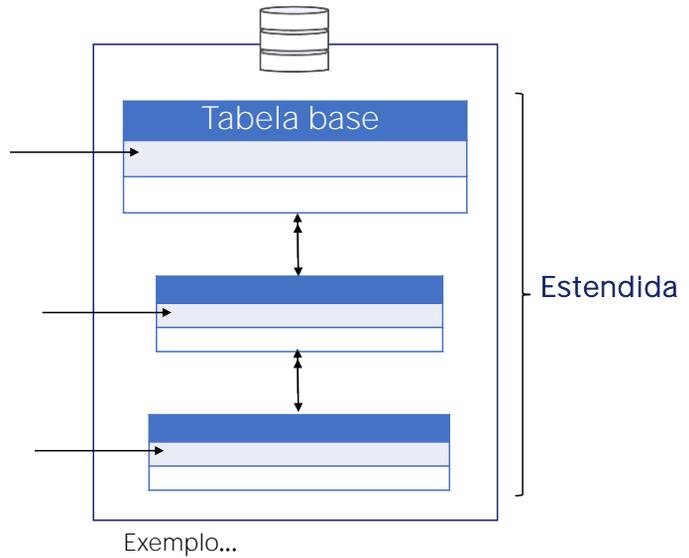
**where** *condition<sub>2</sub>*

...

**where** *condition<sub>n</sub>*

*MainCode*

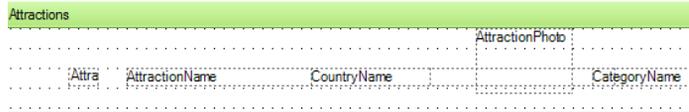
**endfor**



Dentro do comando for each, em seu código principal, os comandos a serem executados são escritos um após o outro, para realizar passo-a-passo o que queremos que aconteça com o registro da tabela base sobre o qual se está posicionado no momento... e os associados pela tabela estendida.

## Sintaxe do For each: Main Code

```
print Title
print ColumnTitles
For each Attraction order CountryName, CategoryName
  print Attractions
endfor
```



### Attractions List

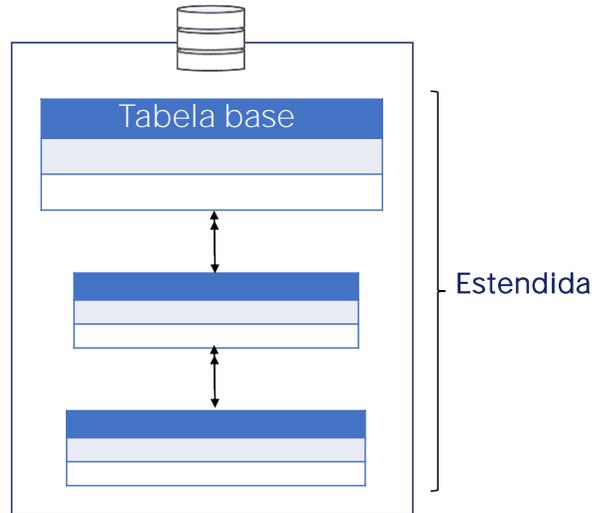
Id	Name	Country	Photo	Category
2	The Great Wall	China		
3	Eiffel Tower	France		Monument
1	Louvre Museum	France		Museum

Por exemplo, imprimir um printblock.

## Sintaxe do For each

```

For each BaseTransaction
  order att1, att2, ..., attn
  where condition1
  where condition2
  ...
  where conditionn
  MainCode
endfor
  
```



Assim fechamos, portanto, a estrutura do comando for each com o que vimos até o momento.

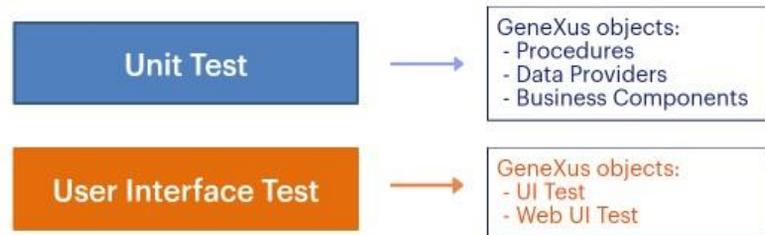
O comando suporta mais cláusulas e opções. Algumas serão vistas em outras aulas. Outras serão abordadas em outros cursos.

# Testes automáticos

Quando desenvolvemos uma nova funcionalidade em nossa aplicação, é necessário testar se o que desenvolvemos funciona conforme o esperado. É importante também testar novamente toda a aplicação após essa alteração, para garantir que o que já estava funcionando continue se comportando corretamente.

À medida que a aplicação cresce, estes tipos de tarefas podem se tornar mais tediosos, pois cada vez mais coisas devem ser testadas, e também mais caras, pois cada vez consomem mais tempo.

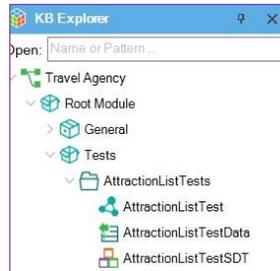
## Automatic Tests



GeneXus nos ajuda dando-nos funcionalidades para poder criar e executar testes automáticos, de modo a poder reduzir parte do trabalho manual de verificação.

- Os **testes Unitários** nos permitem testar uma parte da aplicação de forma isolada. No GeneXus, os testes unitários sem interface se aplicam a testes em procedimentos, DataProviders e BusinessComponents. Resumindo, aqueles componentes onde deveria residir a lógica de negócios de nossa aplicação, e para isso existe o objeto Unit Test.
- O **teste de Interface** –permite criar testes simulando as ações de um usuário, de modo a poder testar fluxos completos da aplicação. Para isso existem os objetos UITest para interface mobile e Web UITest para interface web

## Unit Tests



The screenshot shows the source code editor with the following code:

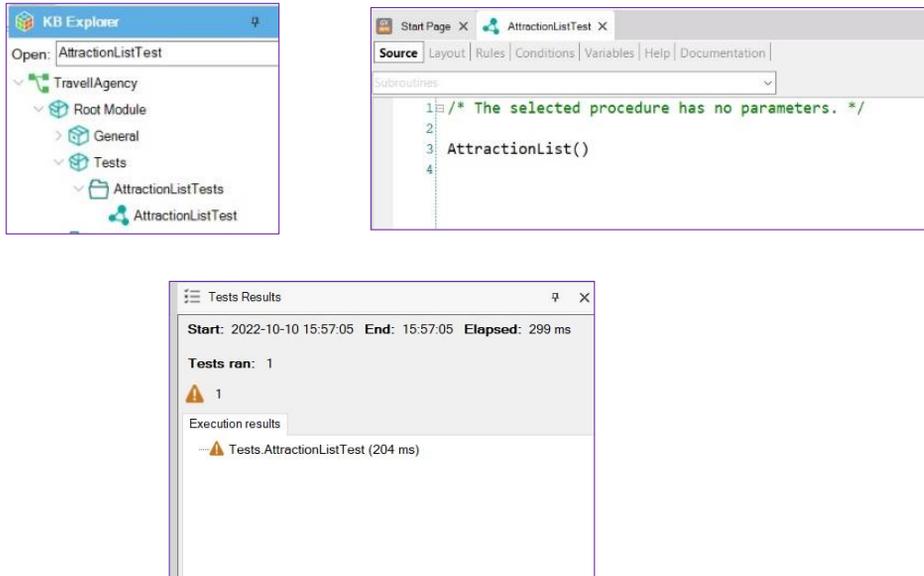
```
1 /* Autogenerated unit test code for Procedure 'AttractionList' */  
2  
3 For &TestCaseData in AttractionListTestData()  
4  
5     /* Act... */  
6     &TestCaseData.Quantity = AttractionList()  
7  
8     /* Assert... */  
9     AssertNumericEquals(&TestCaseData.ExpectedQuantity, &TestCaseData.Quantity, format('!%1.ExpectedQuantity: %2', &TestCaseData.TestCaseId, &TestCaseData.MsgQuant  
10 endfor  
11
```

Vamos nos concentrar nos testes unitários sem interface usando o objeto Unit Test.

Trata-se de um novo tipo de Objeto em nossa KB com a particularidade de ser um Procedimento GeneXus especial. Isto significa que pode ser escrito código GeneXus dentro dele para testar os objetos com diferentes dados e validações.

Vamos ao GeneXus para vê-lo.

## Unit Tests



Vamos criar o correspondente teste que permite testar o procedimento que acabamos de desenvolver.

A maneira mais simples de criá-lo é nos posicionarmos sobre o procedimento AttractionList, clicar com o botão direito e escolher Create Unit Test.

GeneXus cria automaticamente os objetos necessários para implementar o teste automático. Neste caso queremos testar um procedimento que não recebe, processa e retorna algum dado, mas apenas acessa a base de dados para exibir as atrações registradas.

Observemos que na janela KBExplorer é criado um novo nó Tests e, abaixo dele, os testes criados associados aos objetos correspondentes. Para executá-lo, clicamos com o botão direito, Run Test.

GeneXus nos mostrará o resultado de sua execução, tempo decorrido, valores obtidos em diferentes testes, etc.

Embora acabamos de ver um exemplo muito inicial e básico, podemos testar todos os tipos de procedimentos, DataProviders e BC desta maneira. Mais adiante estudaremos estes objetos.

No final do curso e uma vez estudados todos os conceitos necessários, veremos um exemplo mais completo e analisaremos o comportamento do correspondente teste unitário.

*GeneXus*<sup>TM</sup>

[training.genexus.com](http://training.genexus.com)  
[wiki.genexus.com](http://wiki.genexus.com)