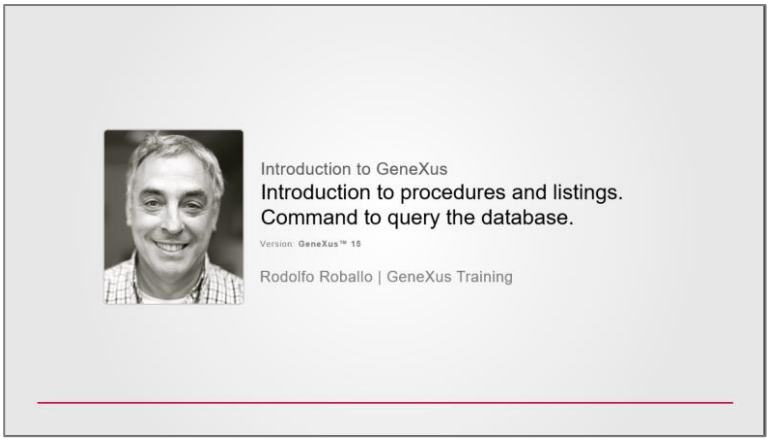
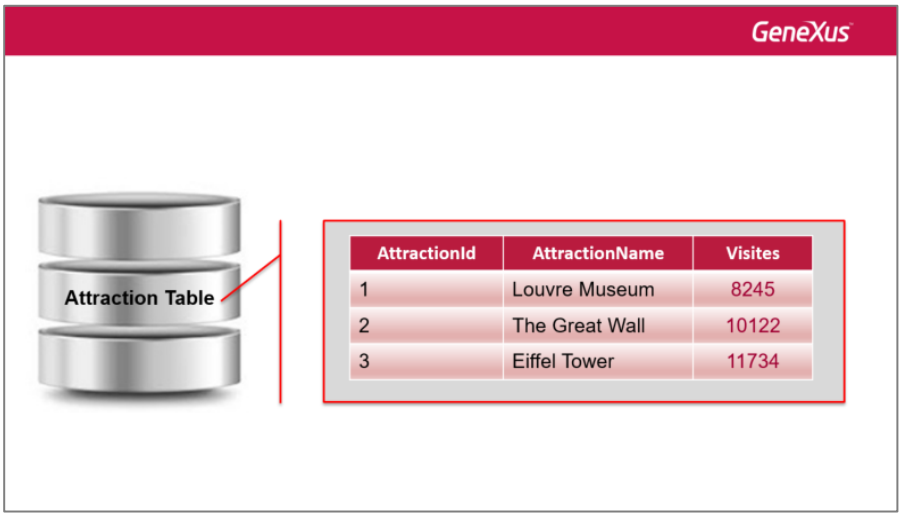
Introdução a procedimentos e relatórios. Comando para consultar o banco de dados.



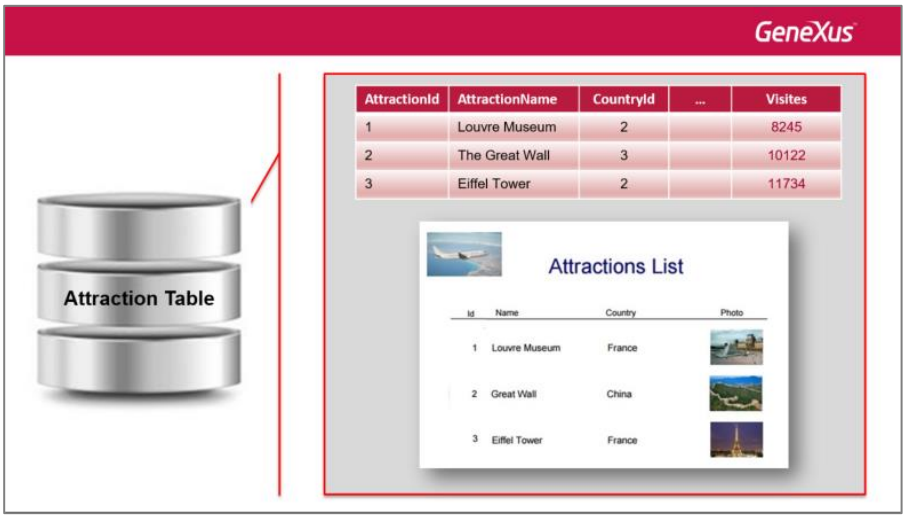
Vamos agora conhecer os objetos do tipo **Procedimento (Procedure)** que podemos definir em uma Base de Conhecimento GeneXus.

Este tipo de objeto nos permite definir **PROCESSOS <u>para acessar e navegar tabelas do banco de dados com diferentes objetivos</u>:**

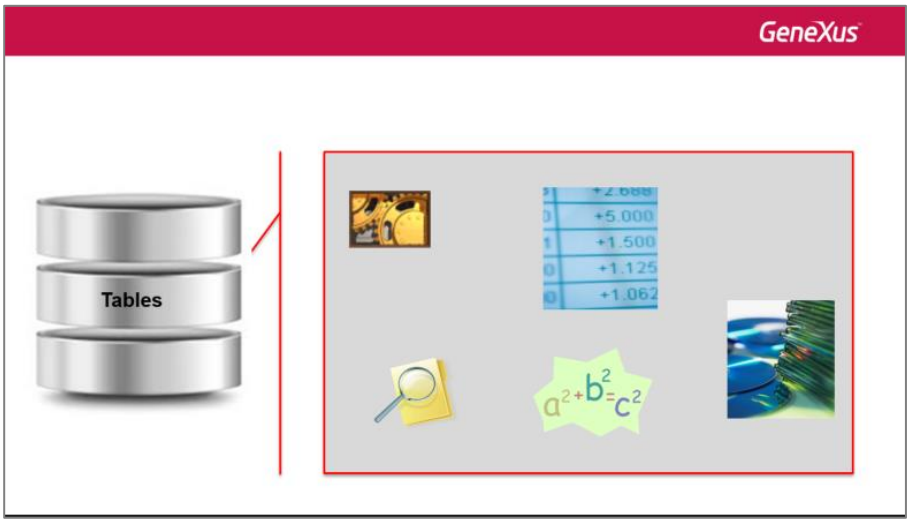
→ por exemplo, podemos precisar navegar pelos registros de determinada tabela, que atendam determinadas condicões, e, nesses mesmos registros, atualizar um de seus atributos com um valor específico.



→ ou navegar determinada tabela para imprimir todos os seus dados em um relatório PDF, ordenados por algum critério.

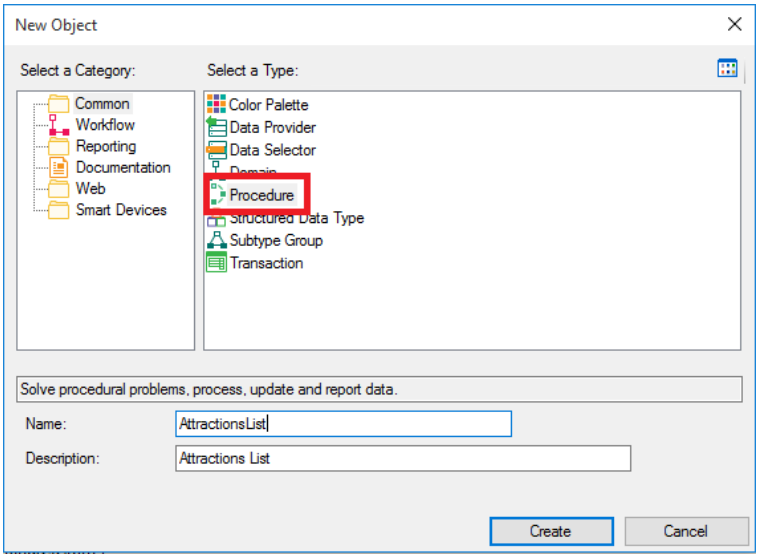


→ ou ainda, definir qualquer processo específico em que necessitamos fazer buscas, cálculos, atualizações no banco de dados ou impressão de informações.

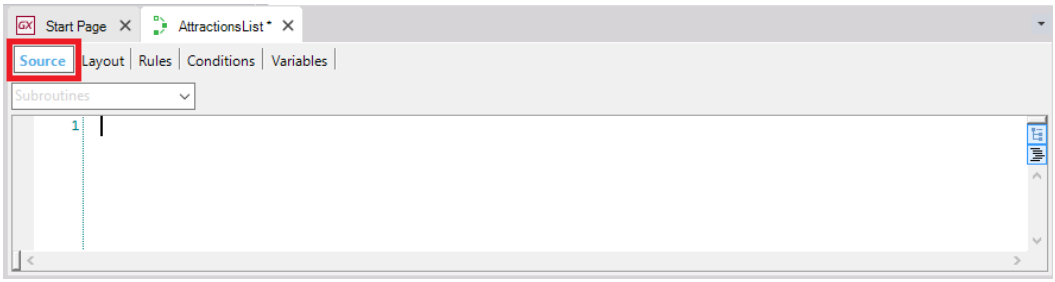


Primeiramente, vamos definir um procedimento para imprimir todas as atrações turísticas da agência de viagens, listadas em ordem alfabética.

Para isso, vamos criar um novo objeto, do tipo **Procedure**, que chamaremos de :

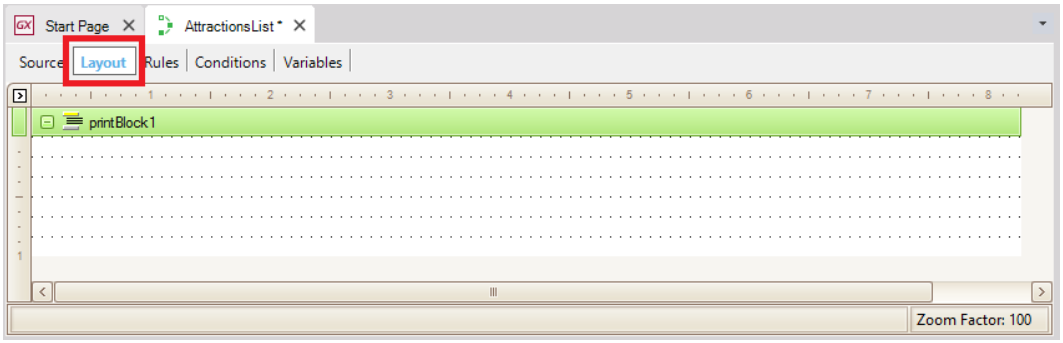


Uma vez criado o objeto, vemos que GeneXus nos posiciona em uma seção chamada **Source**:



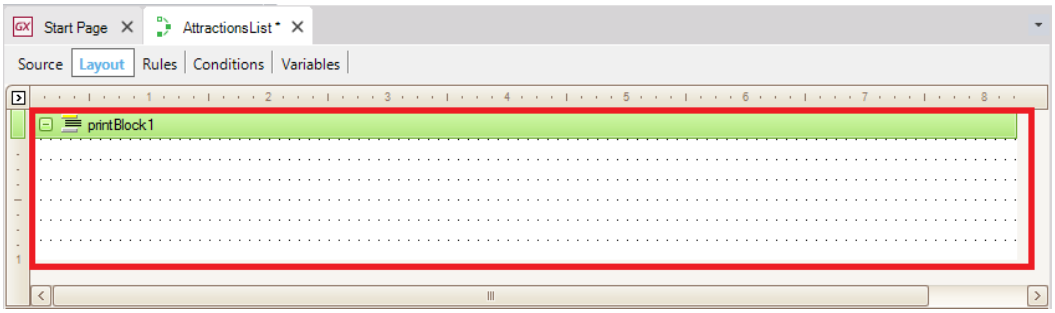
Aqui é onde vamos escrever os comandos e ordens que permitam ao procedimento cumprir com o objetivo para o qual foi criado. Em nosso caso, imprimir uma lista de atrações turísticas.

Agora, observemos esta outra seção chamada **Layout**:



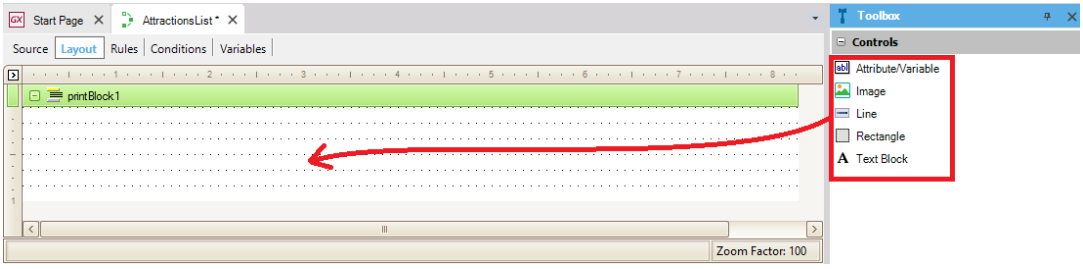
O **Layout** é o lugar destinado para desenhar a saida das informações, ou seja, onde especificaremos como desejamos visualizar nossos dados.

É composto por elementos chamados **printblocks.**



Dentro dos printblocks, incluiremos tudo o que desejamos mostrar.

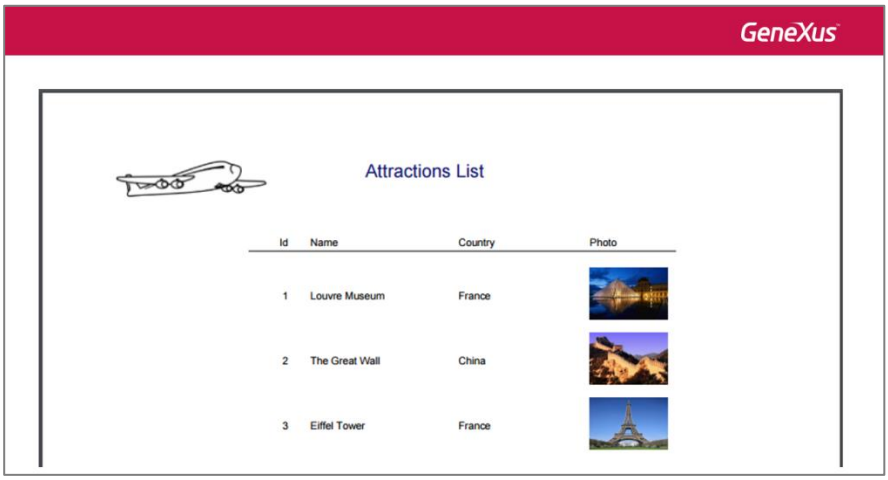
Podemos mostrar títulos, linhas, retângulos, imagens, assim como valores de atributos e variáveis. Para isso, arrastamos algum desses elementos dentro do **printblock**.



Observem que o **Layout** já nos traz um printblock, automaticamente.

Neste printblock, podemos incluir um título, a data atual, etc. Também podemos incluir mais printblocks, como veremos.

Pensemos agora **como queremos visualizar nosso relatório**. Assim, poderemos definir o desenho do layout. Poderia ser algo assim…

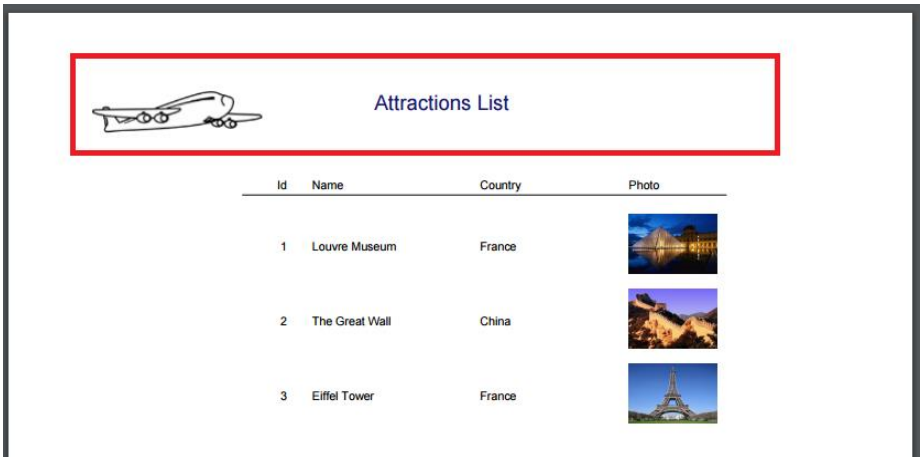


Uma imagen, um título, algumas colunas e todas as atrações turísticas, listadas em ordem alfabética.

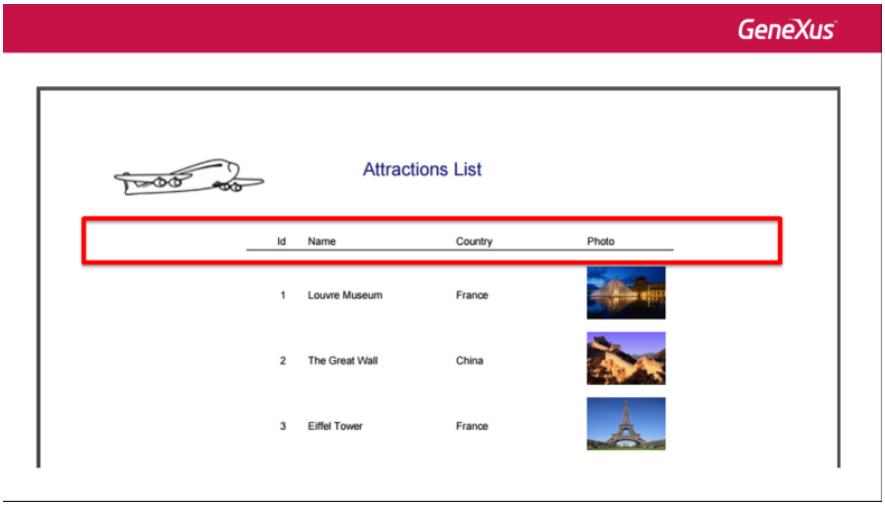
Observem que devemos mostrar 2 conteúdos fixos: o título do relatório com uma imagem e os títulos das colunas, logo abaixo.

Depois, a informação que desejamos ver, ou seja, as atrações turísticas visualizadas linha a linha, já que queremos mostrar todos os dados que temos armazenados em nosso banco de dados.

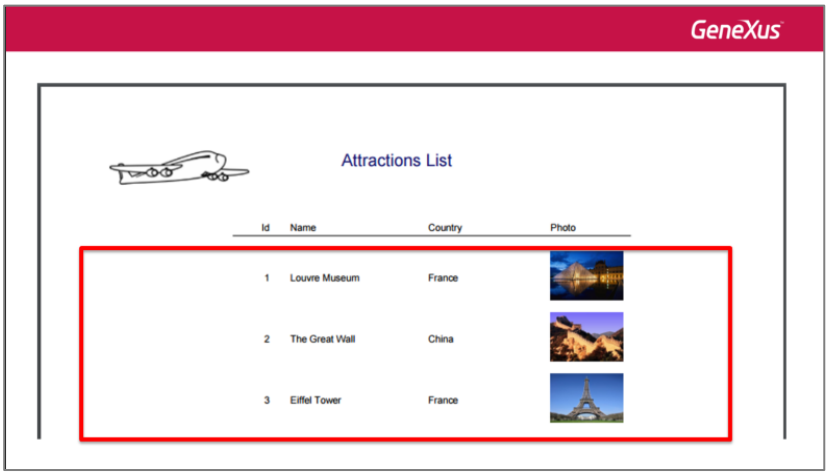
Poderíamos definir, então, três printblocks: no primeiro incluiríamos o título do relatório e a imagen.



No segundo, os títulos das colunas com uma linha embaixo.



e no terceiro printblock

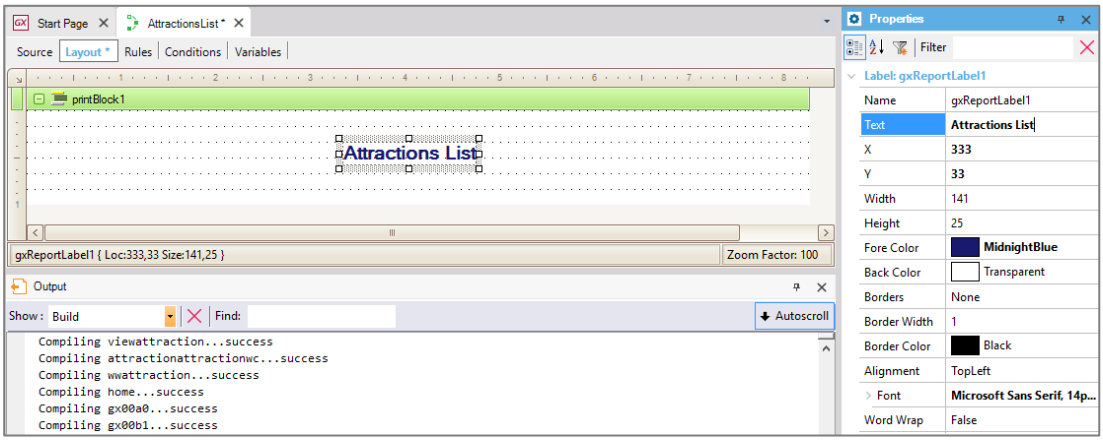


mostraremos os dados das atrações turísticas.

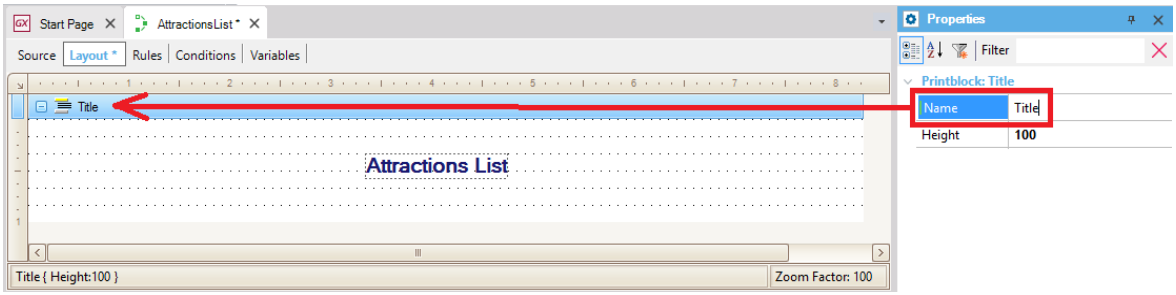
Vamos começar, então, a definir isso.

Podemos usar o printblock que foi criado automaticamente quando criamos o objeto procedimento, para definirmos o título e a imagem.

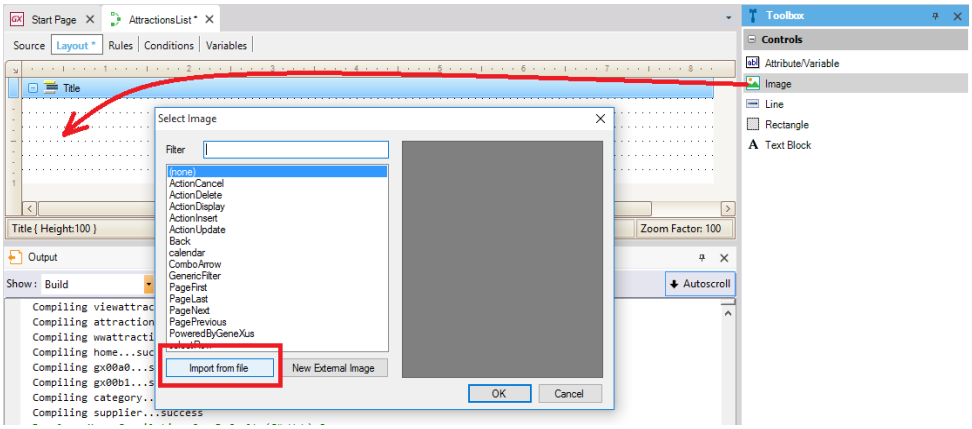
Comecemos pelo título. Para isso, vamos até a Toolbox… Arrastramos um controle Text Block… editamos suas propriedades… e na propriedade Text escrevemos “AtractionsList”. Modificamos também sua cor, sua fonte... e o posicionamos onde queremos que apareça.



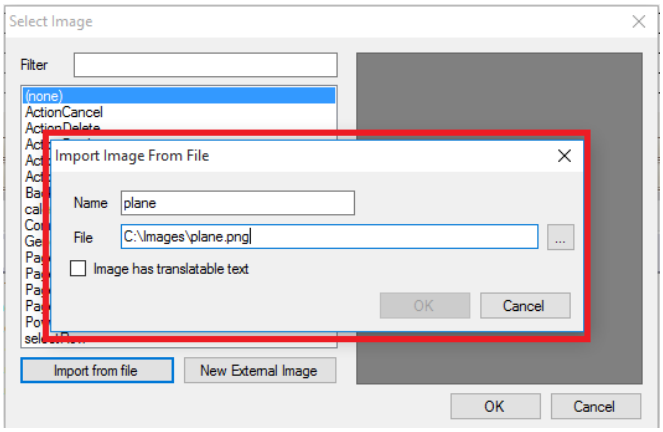
Daremos a este printblock um nome claro, que represente aquilo que está mostrando. Então, acessamos as propiedades do printblock, editamos sua propiedade Name, e mudamos seu valor para “Title”.

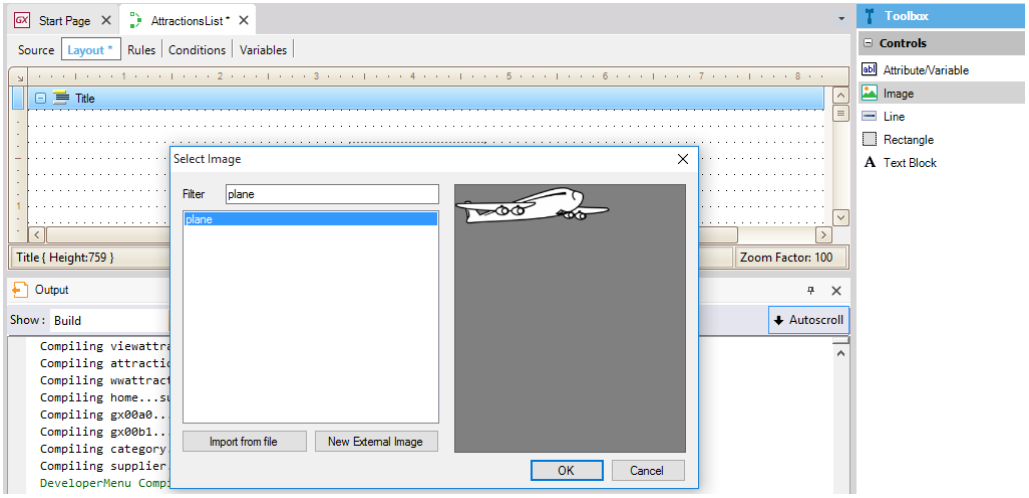


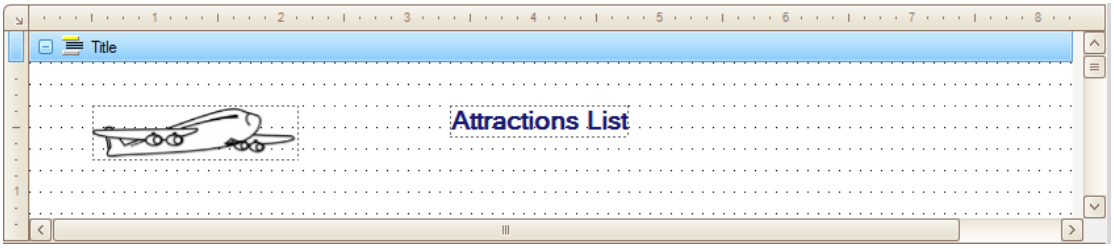
Incluiremos, agora, a imagem com o avião, à esquerda. Para isso, arrastamos da Toolbox um controle Image e o soltamos onde desejamos coloca-lo. Observemos que se abre um popup que nos permite selecionar alguma das imagens existentes na base de conhecimento, ou incorporar uma nova, por exemplo importando-a de um arquivo:



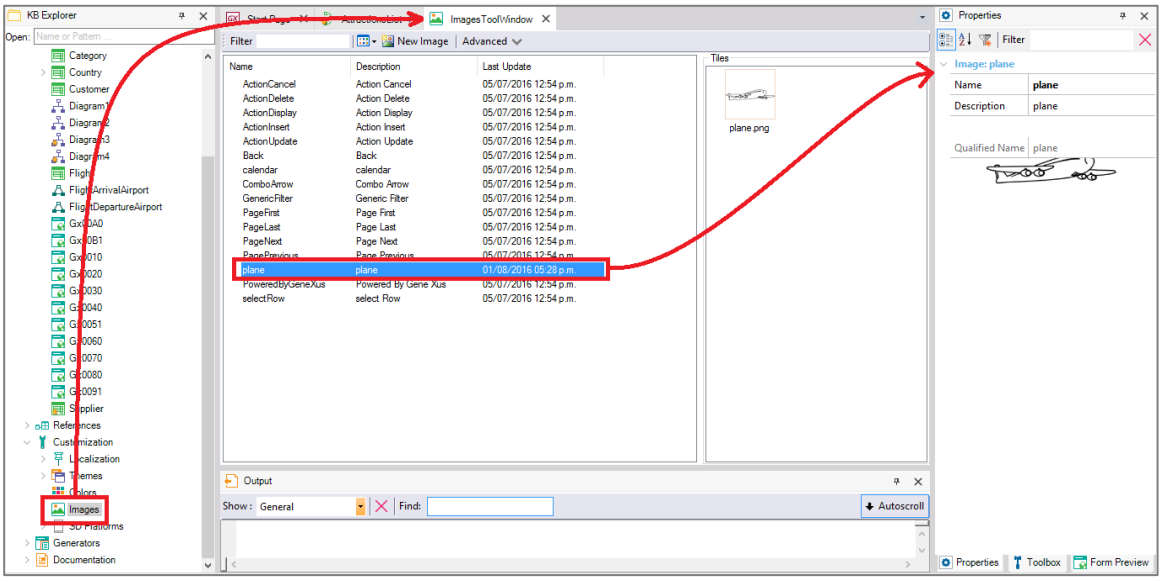
Este botão nos permite explorar nossos arquivos e escolher uma imagem, que será criada como um objeto GeneXus do tipo Image com o mesmo nome do arquivo original. Daí para frente poderemos utilizar essa imagem dentro da nossa KB livremente.



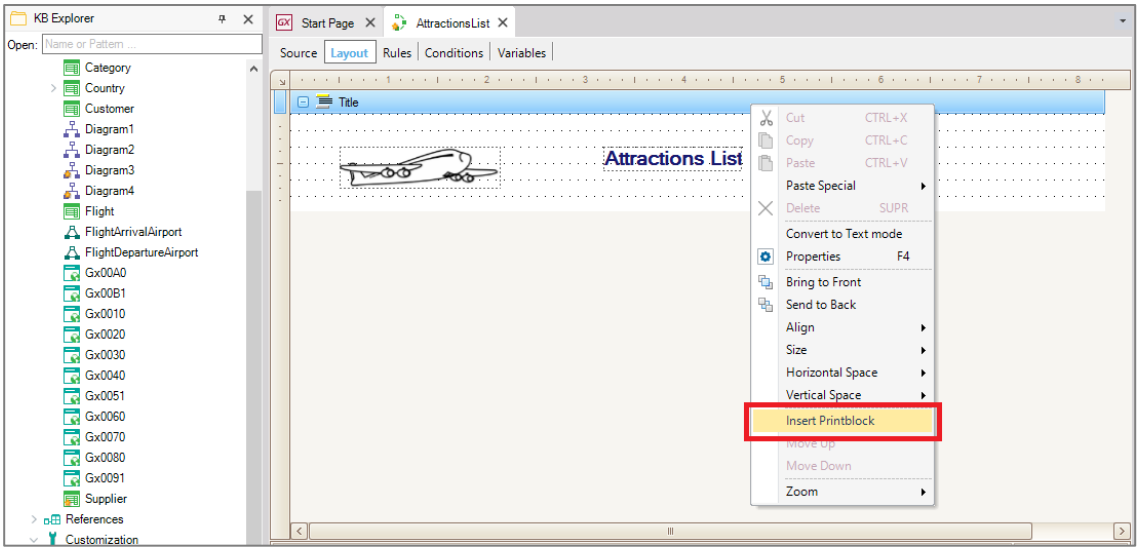




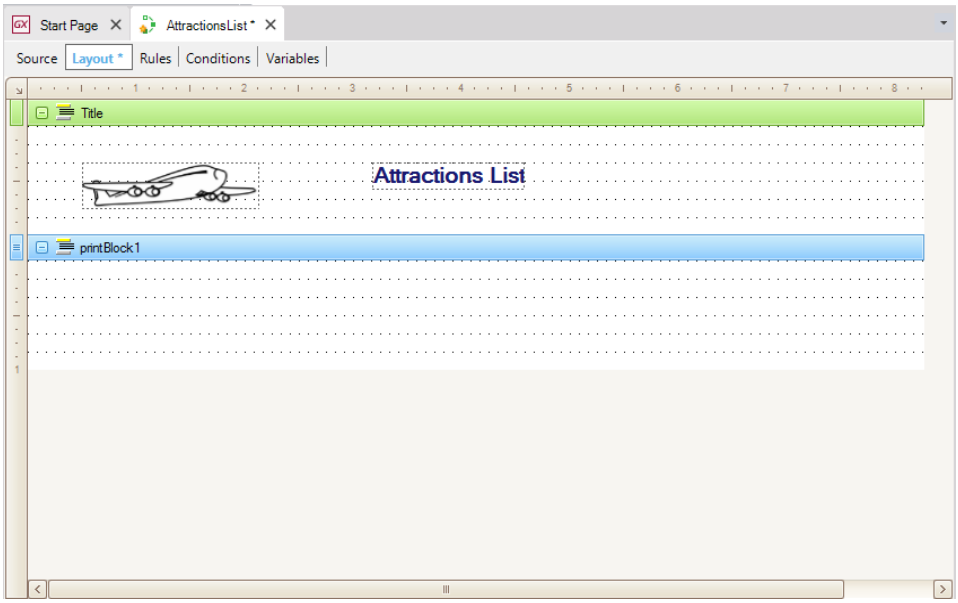
Observemos que aqui podemos ter acesso a todas as imagens da KB, e entre elas está a do nosso avião.



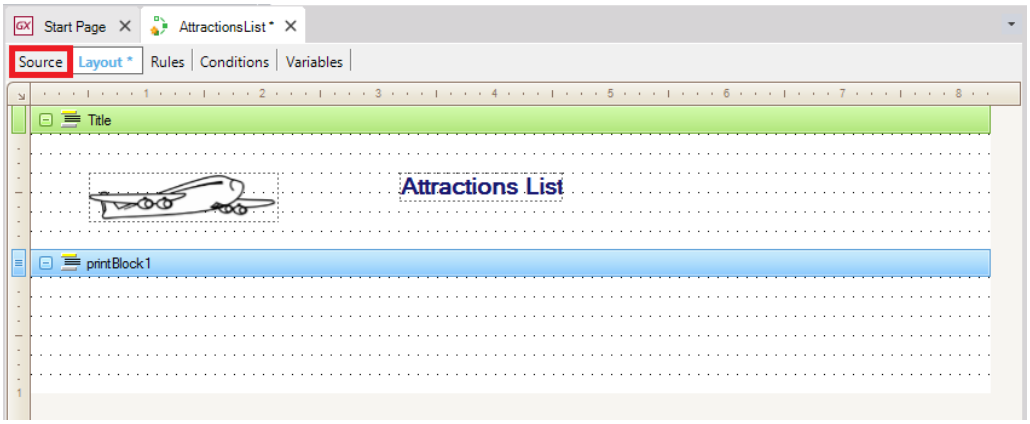
Agora, vamos criar outro printblock para incluir os títulos das colunas, com uma linha embaixo. Se pressionamos o botão direito do mouse **sobre determinado printblock e** selecionamos a opção “Insert Printblock”



será criado un novo printblock **logo abaixo do que clicamos.**

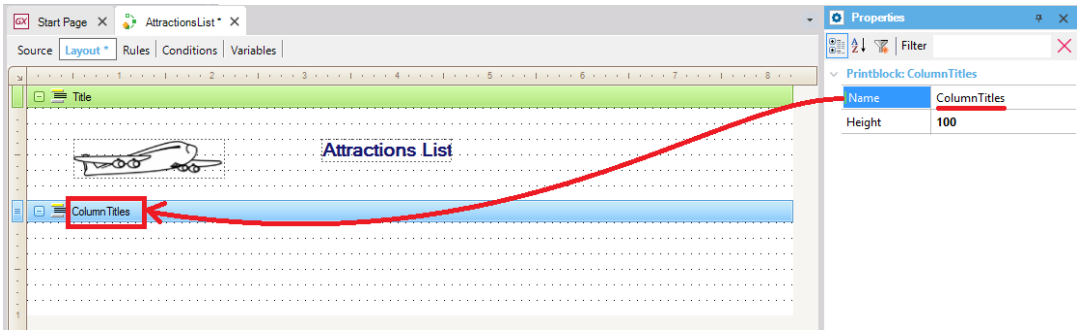


A ordem dos printblocks no Layout não é importante, já que não será necessariamente a ordem em que serão impressos. **Podemos definir quando imprimir cada printblock** no código que vamos escrever no Source do procedimento.



Veremos isto em breve.

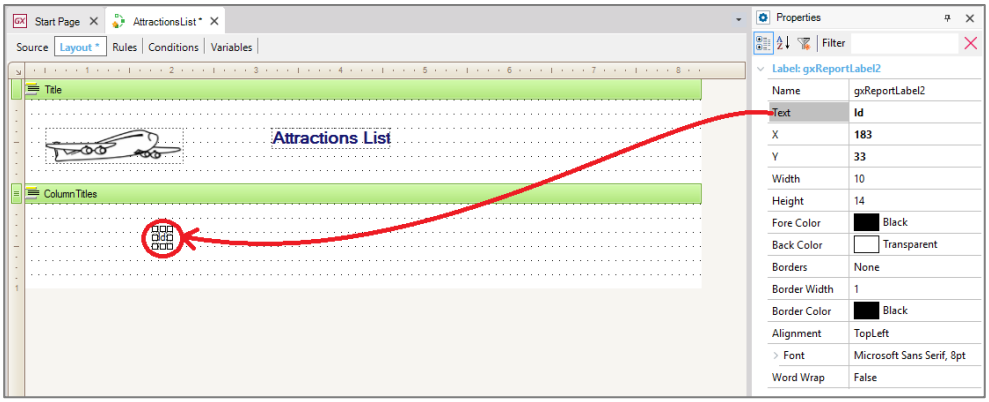
Agora, neste novo printblock, mudaremos seu nome para “ColumnTitles”



… e vamos incluir nele um TextBlock para cada texto que queremos mostrar como título de coluna.

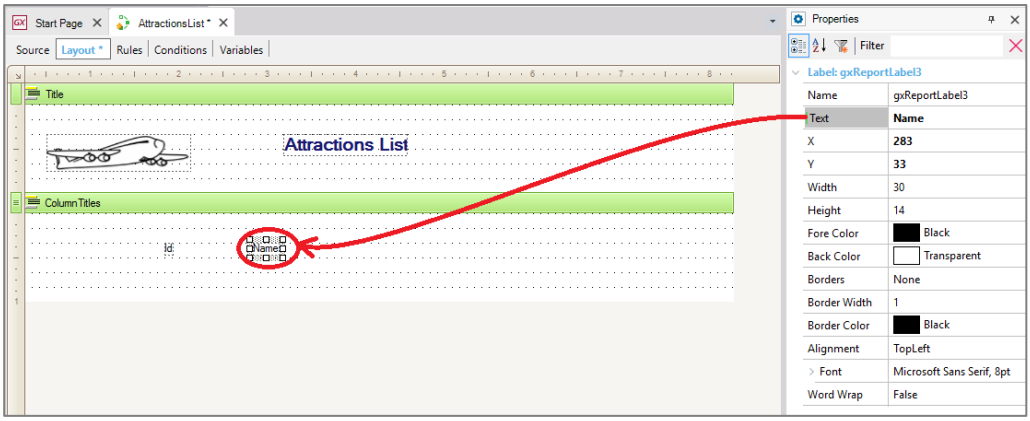
Assim, da Toolbox, arrastramos um Textblock

E, na sua **propiedade** Text, escrevemos “Id”.



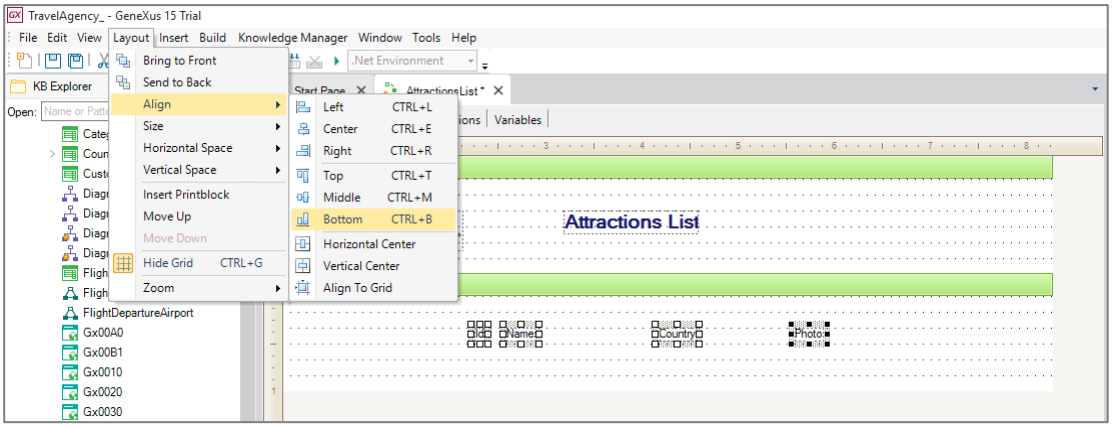
Adicionamos outro Textblock:

E na sua propiedade Text colocamos o valor “Name”:



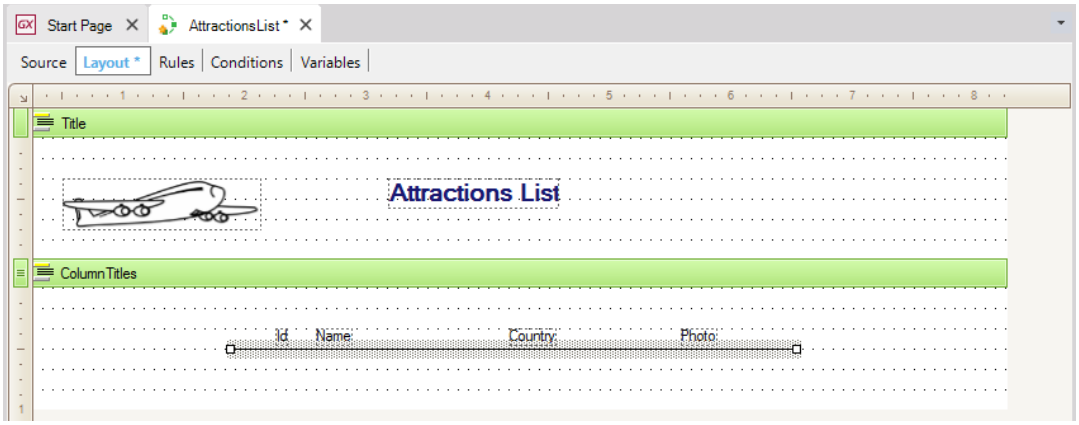
Agora, outro Textblock para mostrar o texto “Country”. E, por úlitmo, um Textblock para o título “Photo”.

Posicionamos os controles onde queremos… Podemos alinha-los, selecionado a todos e então:

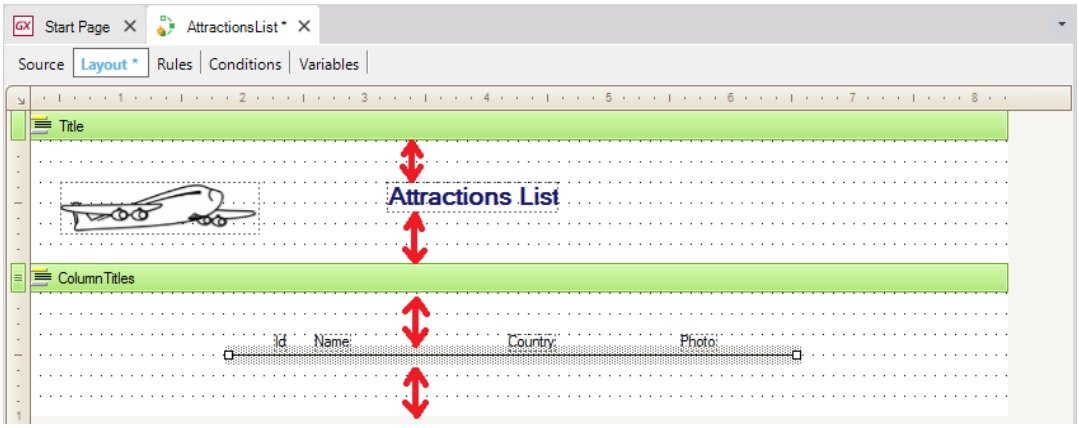


Por último, vamos incluir uma linha debaixo destes títulos de colunas.

Para isso, acessamos a ToolBox, e arrastramos um controle “Line”.
Arrastamos de um ponto inicial… até que se torne uma linha do tamanho que desejamos...

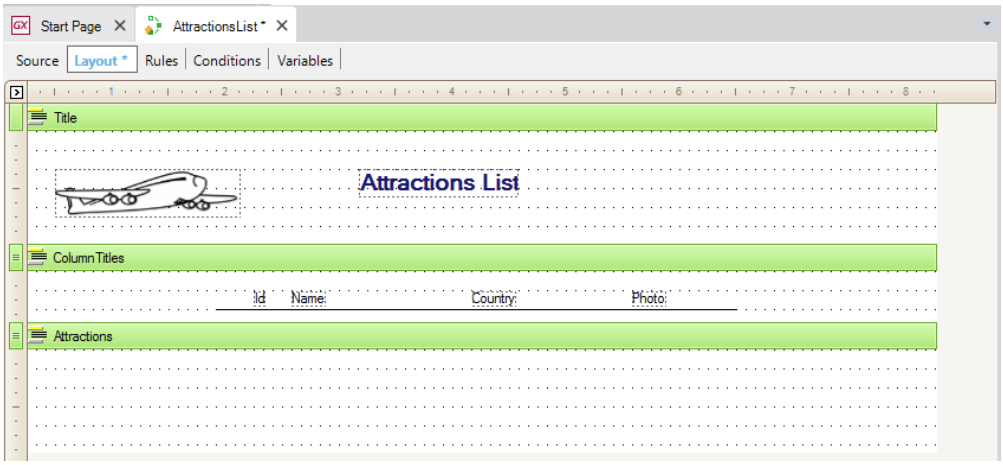


Se agora, no Source, escrevessemos os comandos para imprimir o primeiro printblock e, logo em seguida, o segundo, estes espaços em branco mostrados abaixo seriam mantidos:



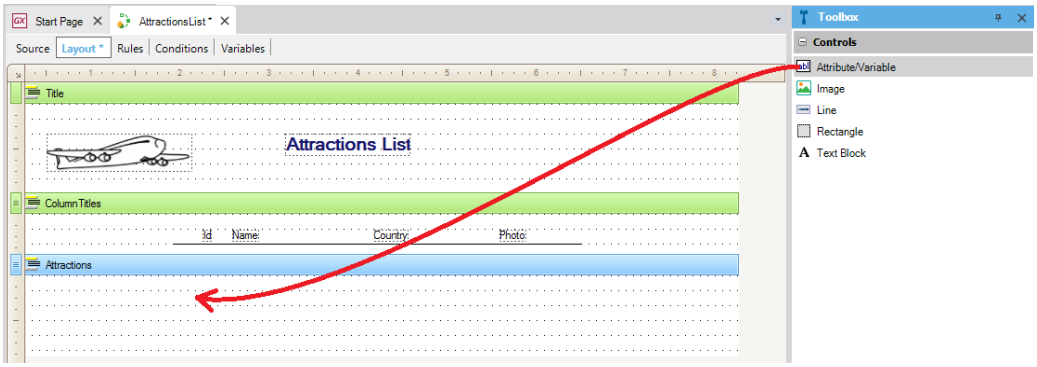
Vamos reduzir os espaços do segundo printblock:

Está faltando, agora adicionarmos o terceiro printblock que havíamos falado, para mostrar os dados das atrações turísticas. Então, adicionamos um novo printblock… e o chamamos de “Attractions”.

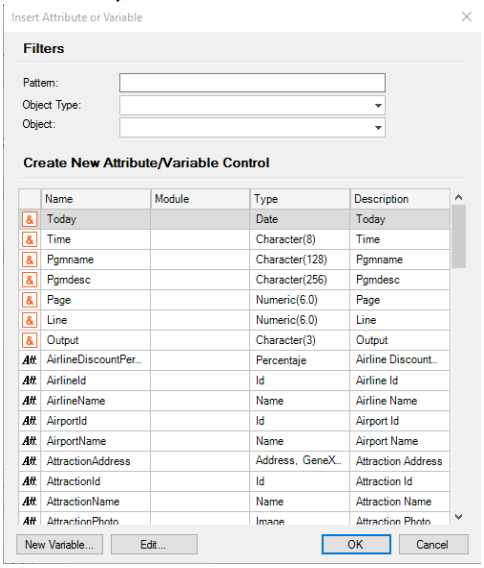


Como os dados estão armazenados em atributos, vamos novamente até a Toolbox, selecionamos um controle do tipo “Attribute/Variable”,

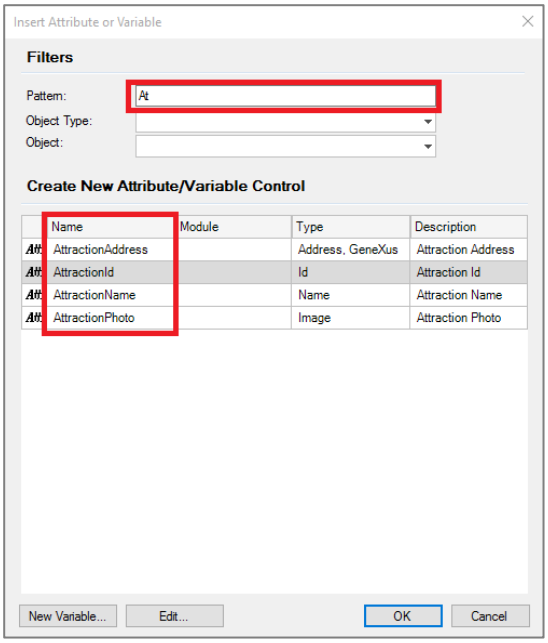
e o arrastramos abaixo do título “Id”…



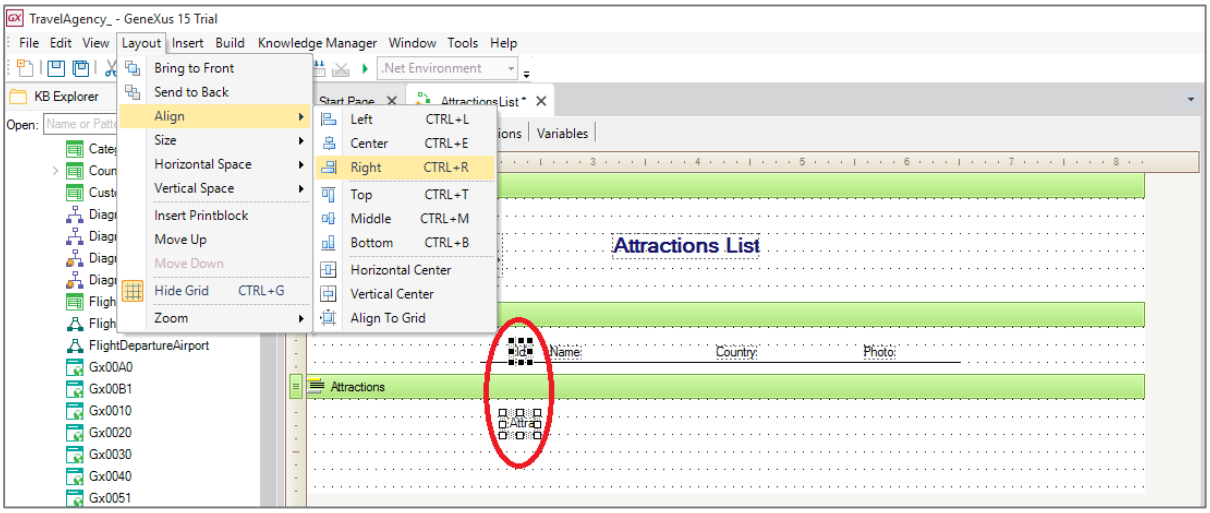
Nesta janela que se abriu, escolhemos qual variável ou atributo queremos mostrar no controle. Vemos que além de &today, existem outras variáveis de sistema que podemos utilizar num procedimento.



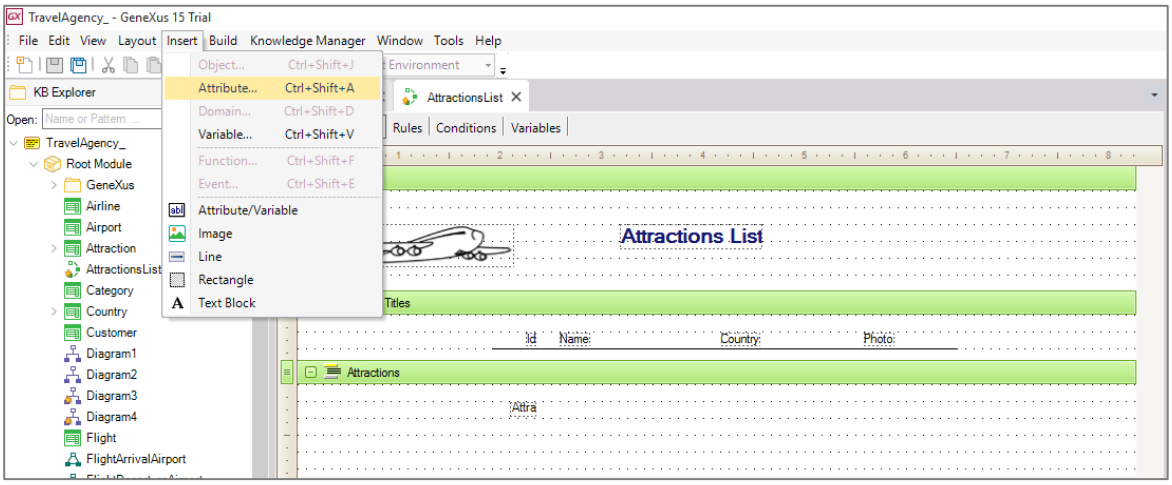
…queremos que mostre o atributo “AttractionId”. Filtramos por “At” e, assim, o encontramos com facilidade:



Alinhamos respectivamente à direita do título “Id”… (O último controle selecionado é o que serve de referência: vemos que este aparece destacado graficamente)

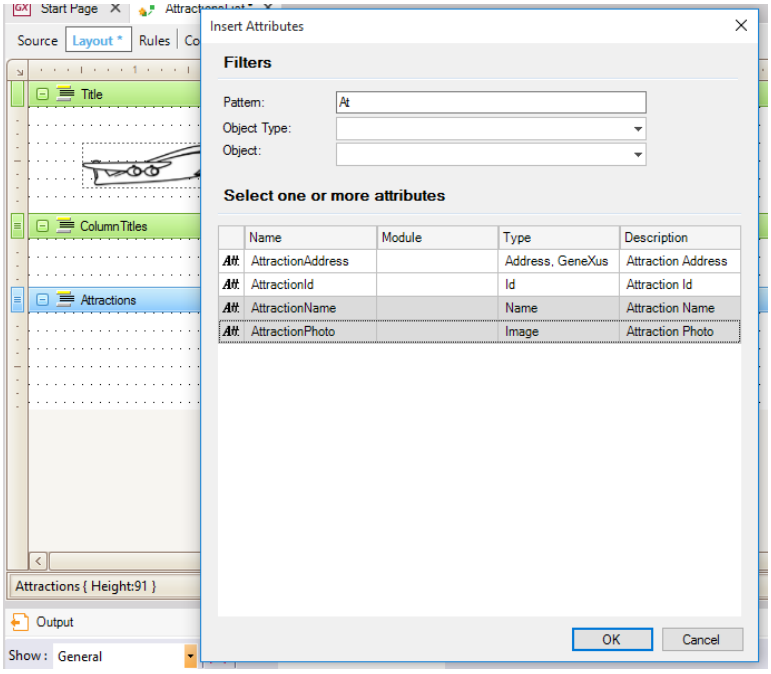


Também podemos incluir atributos em um printblock através do menu **Insert / Attribute**



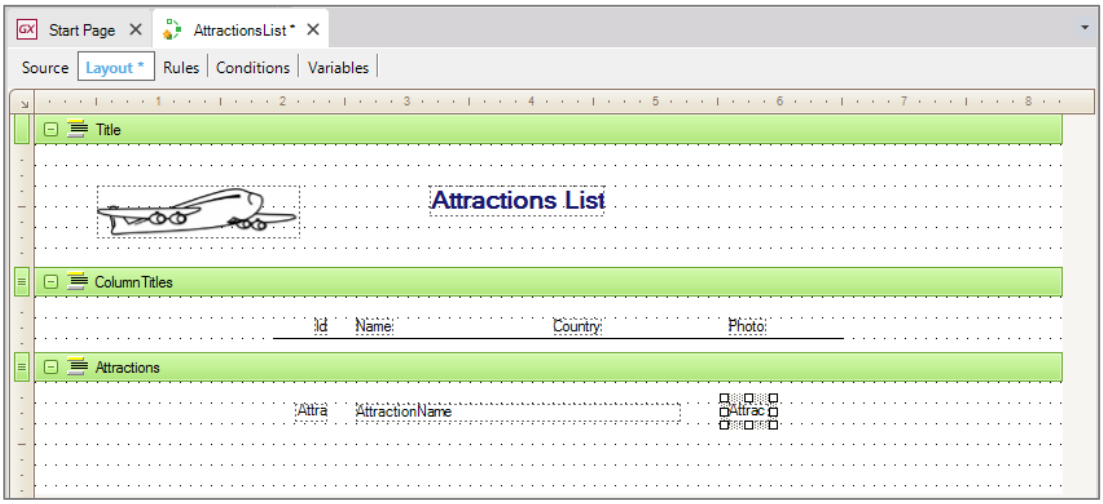
Aqui aparecem somente atributos, e não variáveis. Neste quadro, podemos escolher vários atributos de uma só vez.

Depois que selecionamos AttractionName, pressionamos a tecla Control, e clicamos também em AttractionPhoto.

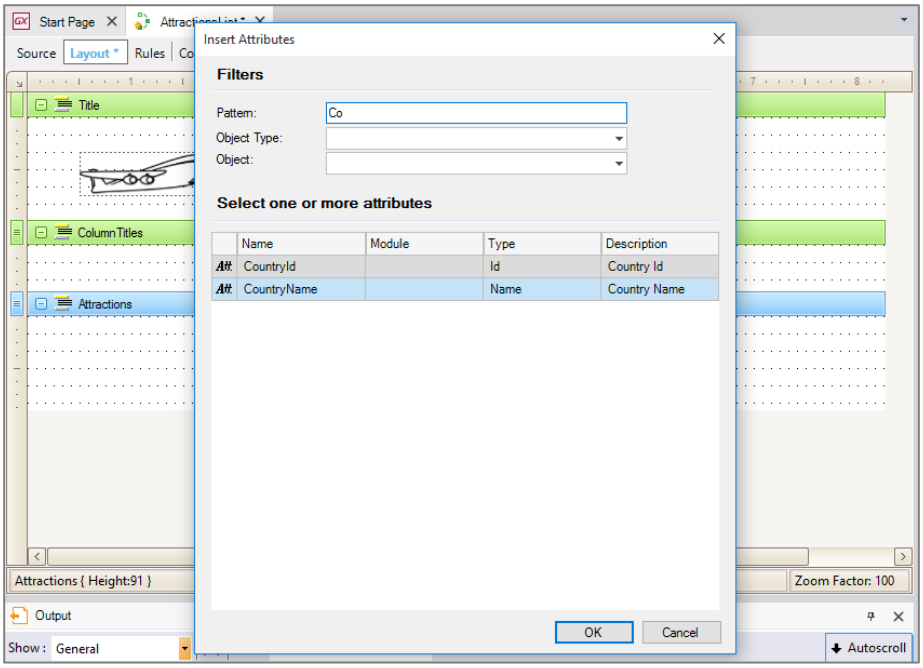


Pressionamos OK...

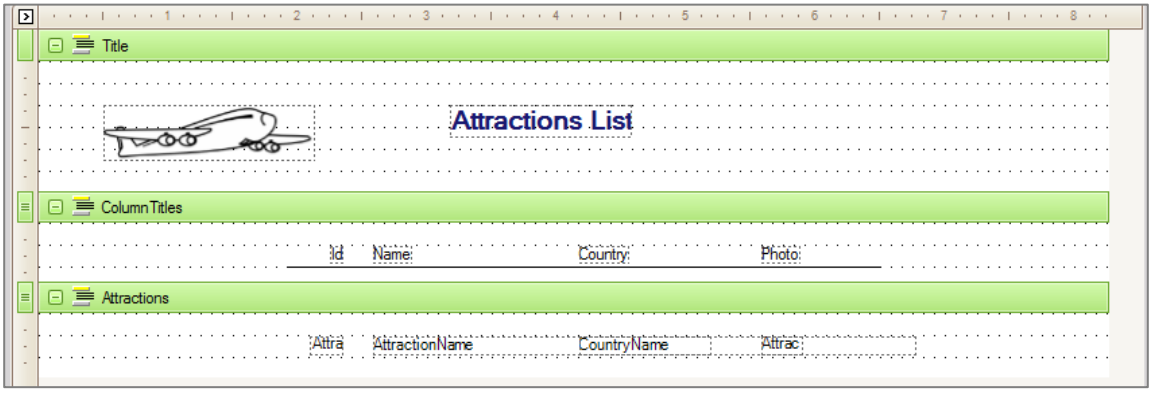
Posicionamos os atributos debaixo dos títulos...



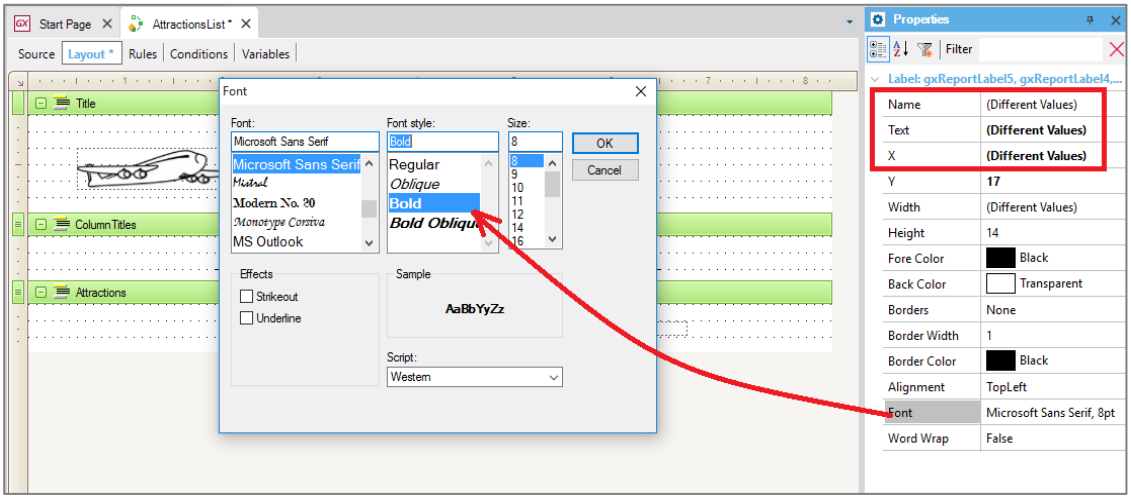
E incluimos CountryName, da mesma forma:

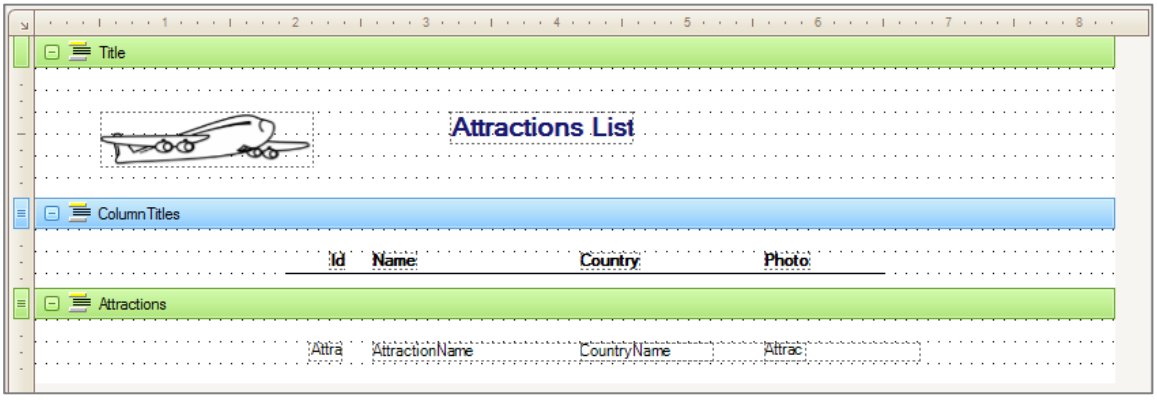


Alinhamos os atributos, e removemos os espaços em branco do printblock.

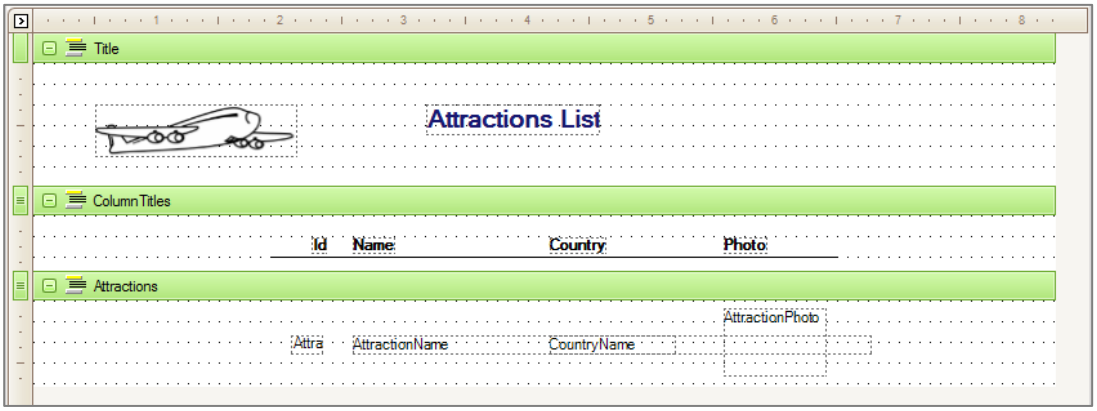


Vamos colocar em negrito os títulos das colunas, para que se destaquem. Selecionamos a todos de uma só vez, e nas propiedades, vemos que podemos aplicar a mudança na Fonte para todos.





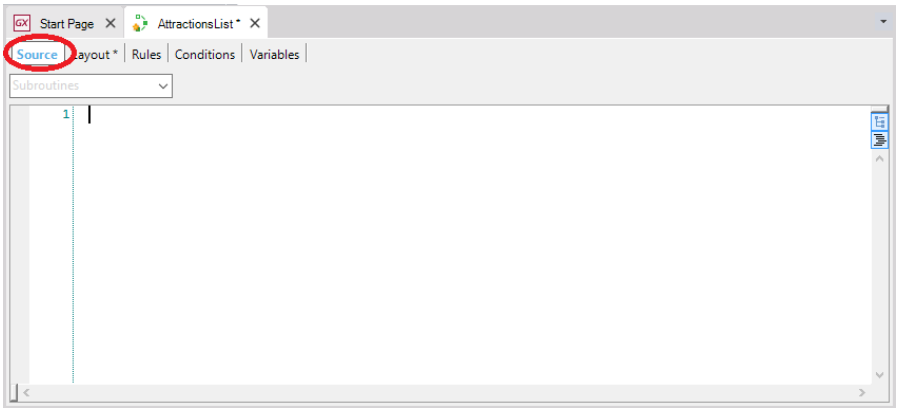
Não nos demos conta de uma coisa: a foto da atração não poderá aparecer nesse espaço tão reduzido que deixamos para o atributo AttractionPhoto. Teremos que aumentá-lo um pouco.



Já temos pronto o desenho de como queremos mostrar a informação no relatório.

Agora falta escrever o código necessário para obter a informação correta do banco de dados e indicar que se imprimam os printblocks na sequência que desejamos.

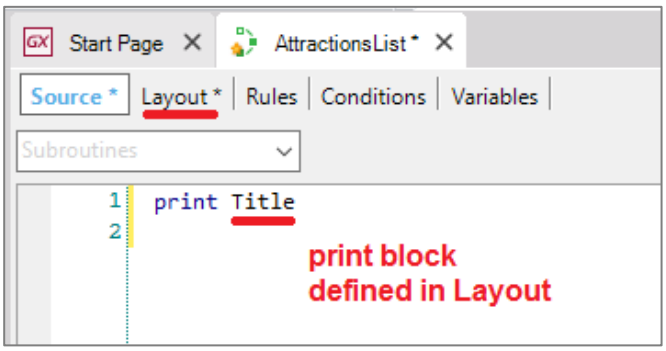
Vamos, então, à opção Source...



A primeira coisa que queremos imprimir é o título do relatório. Então escrevemos “print Title”:

Como as instruções que escrevemos no Source são disparadas de cima para baixo, **esta instrução será a primeira a ser executada**. Estamos indicando com ela, que se imprima o conteúdo do printblock de nome Title, ou seja, o título do relatório.

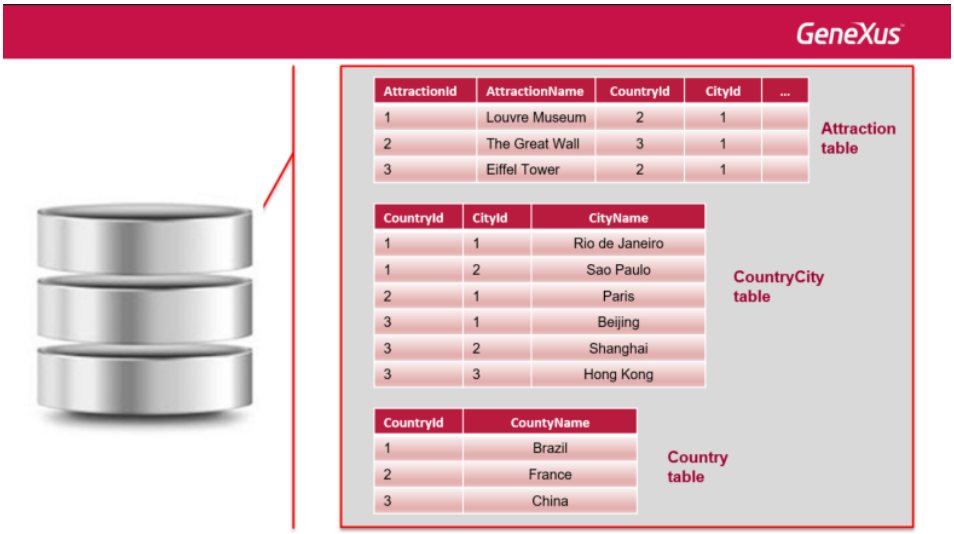
O comando Print sempre deve vir acompanhado do nome de um printblock definido no Layout.



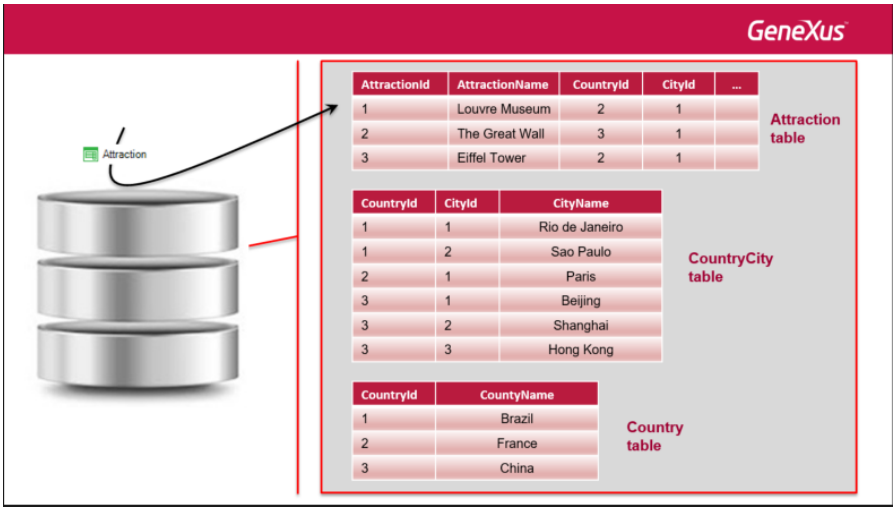
Em seguida, desejamos imprimir os títulos das colunas. Então, temos que dar o comando para imprimir o printblock “ColumnTitles”…
Então, escrevemos “print ColumnTitles”:

Com estas duas instruções, indicamos que se imprima a parte fixa do relatório, ou seja, a que não mudará de acordo com os dados: a que contém o título do relatório e a imagem do avião e a que contém os títulos das colunas.

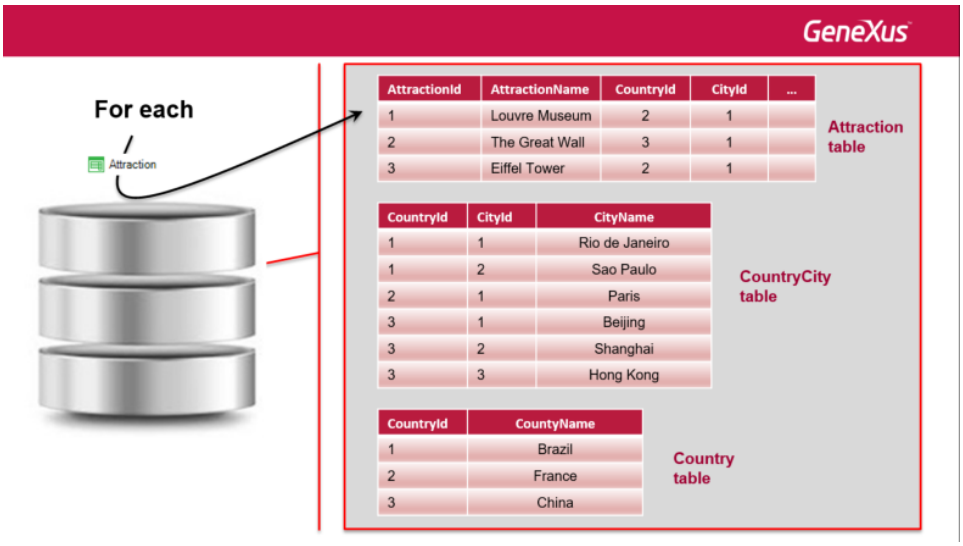
Porém, agora precisamos imprimir as informações das atrações turísticas, que estão armazenadas no banco de dados.



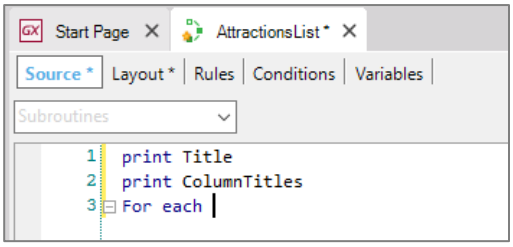
Para isso, devemos acessar a tabela física que tem armazenada esta informação, ou seja, a tabela associada à transação Attraction.



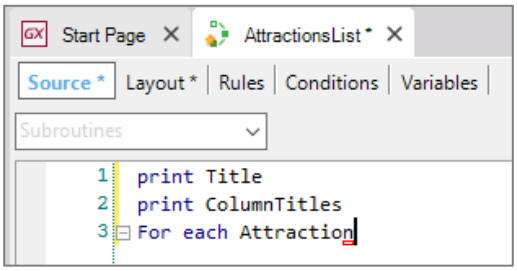
O comando que nos permite acessar uma tabela física é o comando "For Each". A tabela física acessada é chamada de tabela base do for each.



Escrevemos, então, o comando For Each…

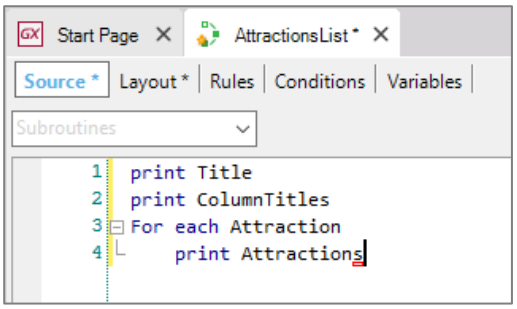


e ao lado: Attraction.

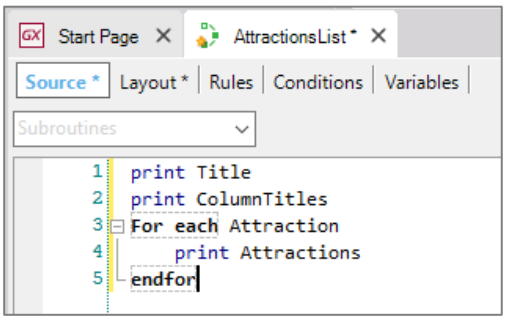


Por que escrevemos Attraction ao lado do For each?
porque é o nome da transação **cuja tabela física associada** queremos navegar…

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

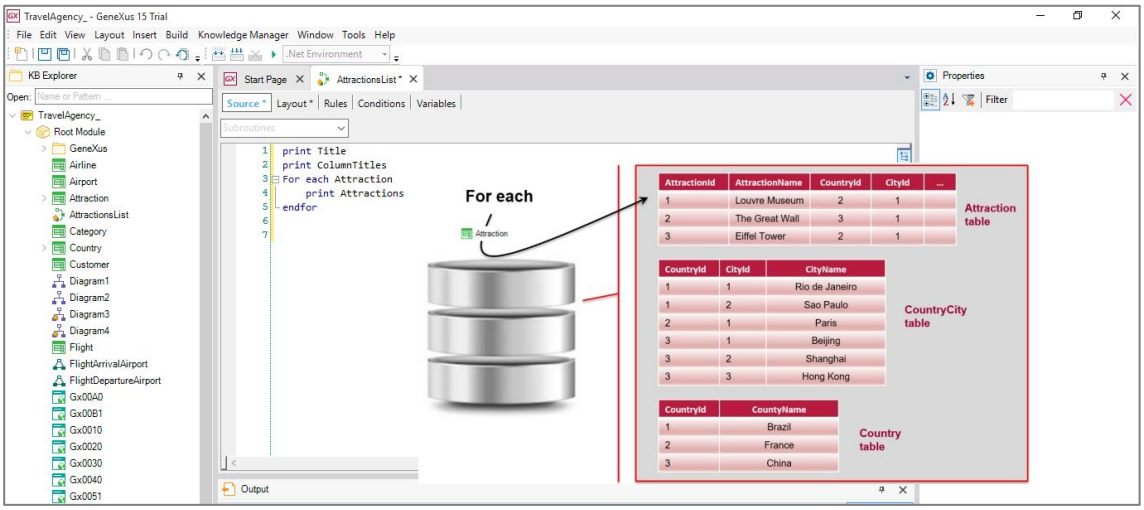


e em seguida fechamos o comando For Each con a instrução Endfor

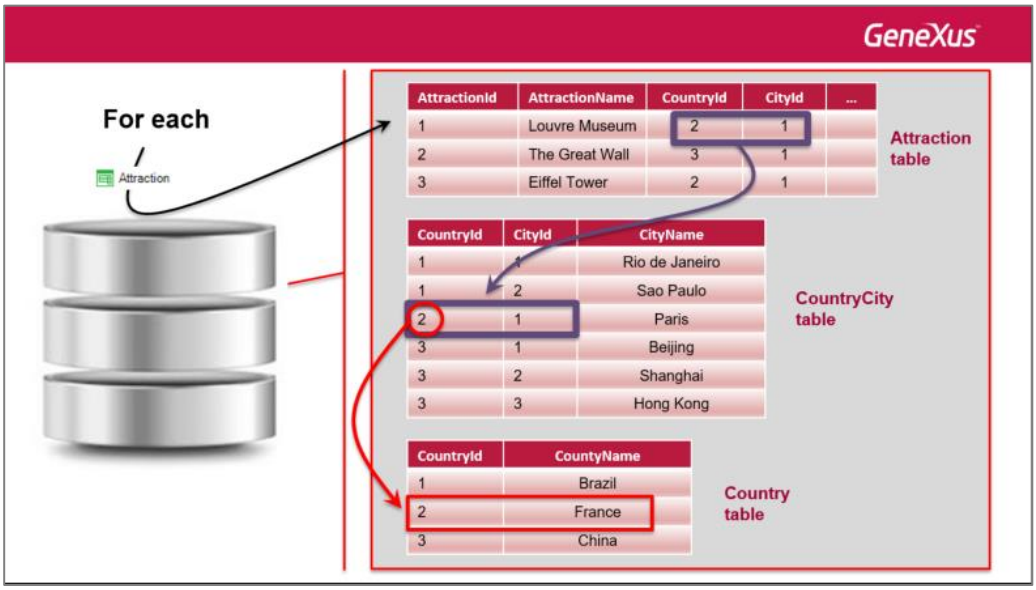


Pronto!

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

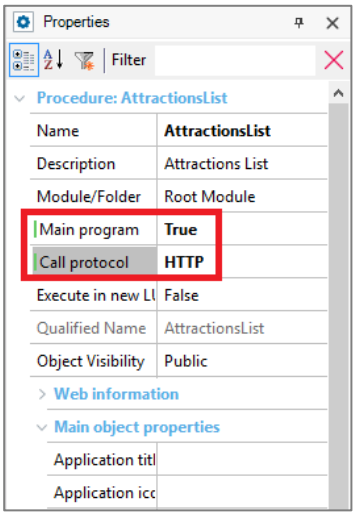


e como dentro do For each, invocamos um printblock que contém atributos das tabelas ATTRACTION e COUNTRY, aplicando o conceito de tabela extendida, para cada atração navegada, acessaremos a tabela COUNTRYCITY, e através desta, chegamos a COUNTRY, para obter o nome do país onde se encontra cada atração:

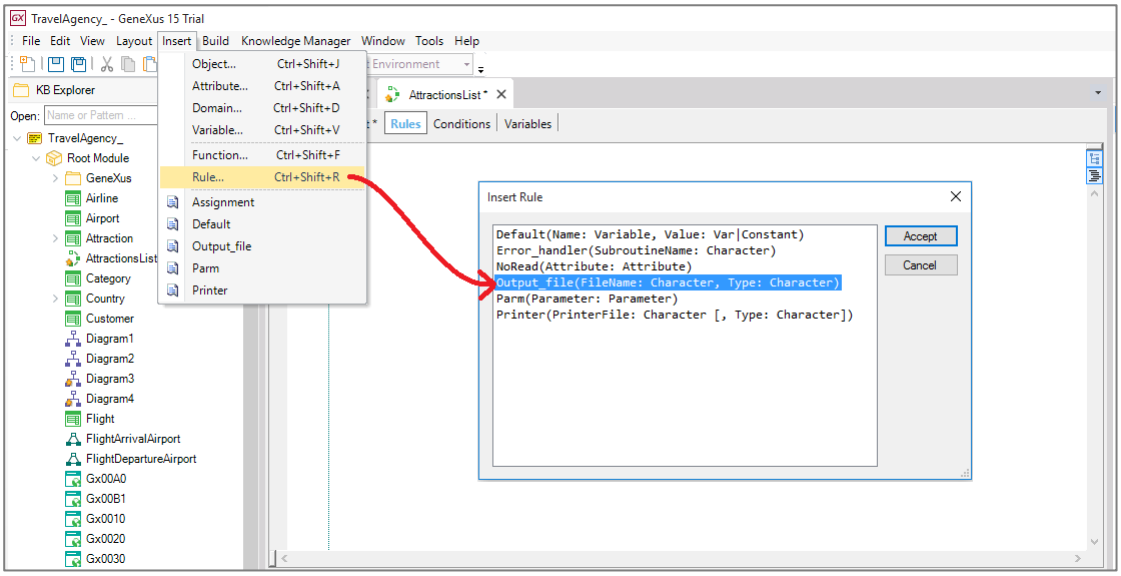


Vamos executar para ver o resultado.

Porém, primeiro temos que definir algumas propiedades necessárias para que se imprima o relatório com formato PDF. (para facilitar a visualização das propriedades use a visão categorizada, não alfabética) Vamos às propiedades do relatório e na propiedade “Main program” selecionamos True.
Depois, na propiedade “Call protocol” selecionamos “HTTP”.

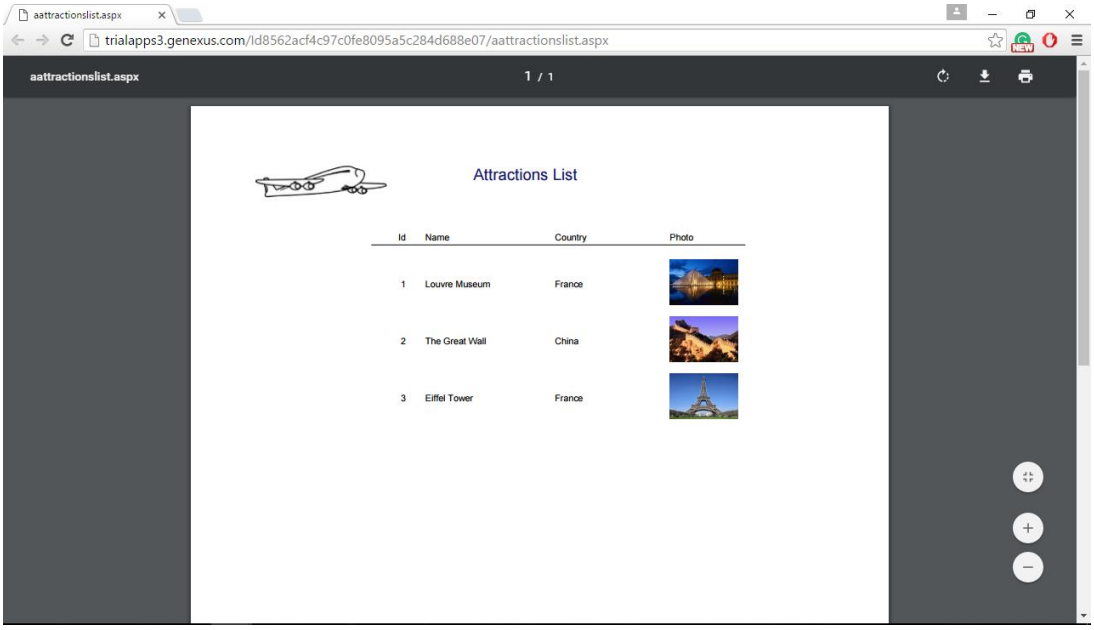


Por último, temos que incluir a regra OutputFile na sección de Rules… porque, como podemos ver, um objeto deste tipo também permite definir algumas regras – embora menos que numa transação–. Então, selecionamos Insert/Rule.



Completamos colocando o nome do archivo do relatório “AttractionsList.PDF” e em seguida o formato que vamos utilizar: “PDF”.

Salvamos… agora já podemos executá-lo.
E vemos que o relatório foi gerado! … com o formato que definimos… e aparecem listadas todas as atrações turísticas que havíamos cadastrado, cada uma delas com o nome do país a que pertenece e com sua foto.



Voltemos ao GeneXus

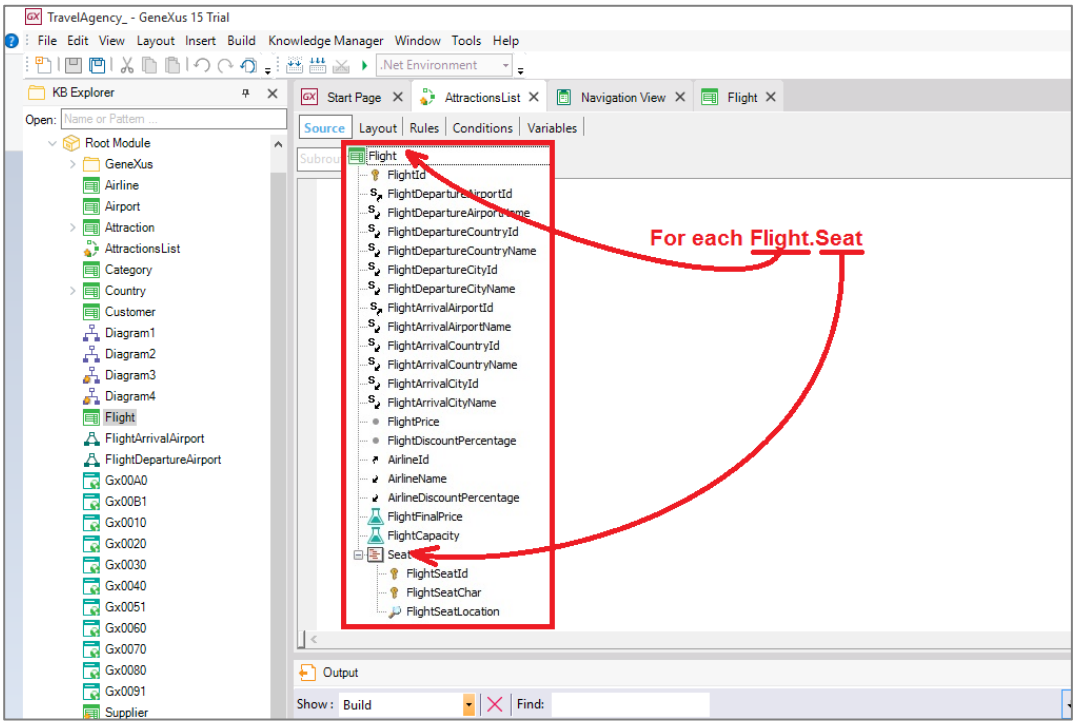
Aqui denominamos a Attraction como: **Base Transaction**

For each Attraction → BASE TRANSACTION
  Print Attractions
Endfor

e como dissemos antes, corresponde ao nome da transação, cuja tabela física associada queremos navegar. No nosso caso, trata-se da tabela Attraction, com o mesmo nome da transação.

E se a transação tivesse mais de um nível? Como Flight, por exemplo... e quiséssemos navegar a tabela física asociada ao 2do nível da transação? Quer dizer: aos assentos do vôo.

A sintaxe, neste caso, seria a seguinte:



Ou seja: o nome da transação:

For each Flight.Seat

Transaction name

Digitamos ponto

E depois do ponto, o nome do nível:

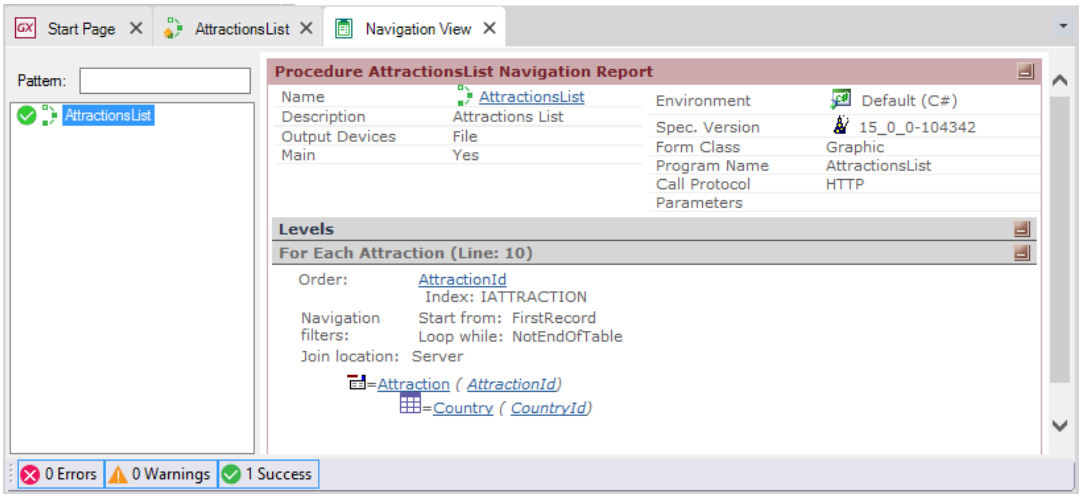
For each Flight.Seat

Transaction name | Transaction Level name

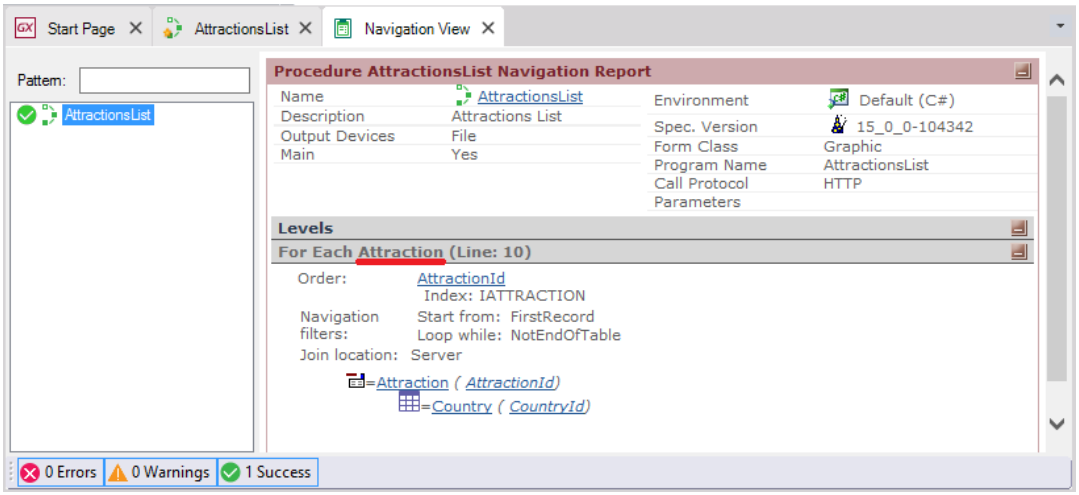
Bom. **A tabela física que o For each navegará**, assim como outras decisões que o Genexus toma, são mostradas na **lista de navegação do procedimento.**

Esta lista é criada automaticamente durante a geração do procedimento antes de ser executado. Em nosso caso, logo depois que pressionamos F5.

No GeneXus, é mostrado como este acessa a informação do banco de dados. Vamos observar:

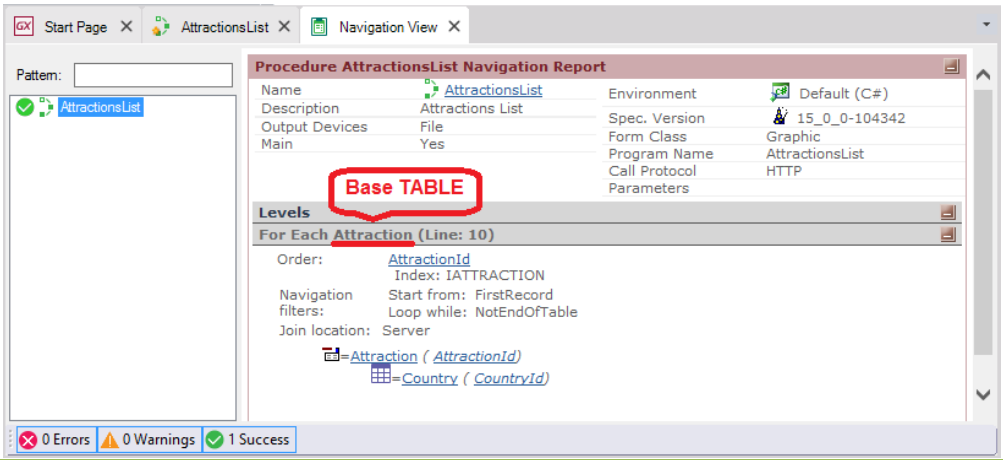


Podemos ver que do lado de onde diz “For Each”, está escrito Attraction:

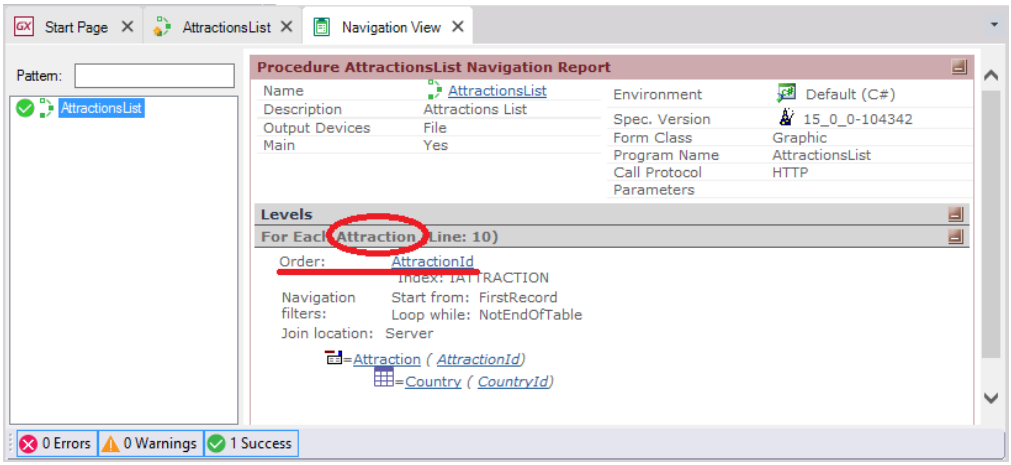


para nos indicar que esta é a **tabela base do For Each.**

Recordemos que o For each percorre uma tabela física, por isso o nome Attraction que aparece na lista de navegação é o da tabela física ATTRACTION, e não o da transação base que tínhamos escrito no procedimento. GeneXus utiliza esta tabela porque é a tabela associada à transação base que indicamos.

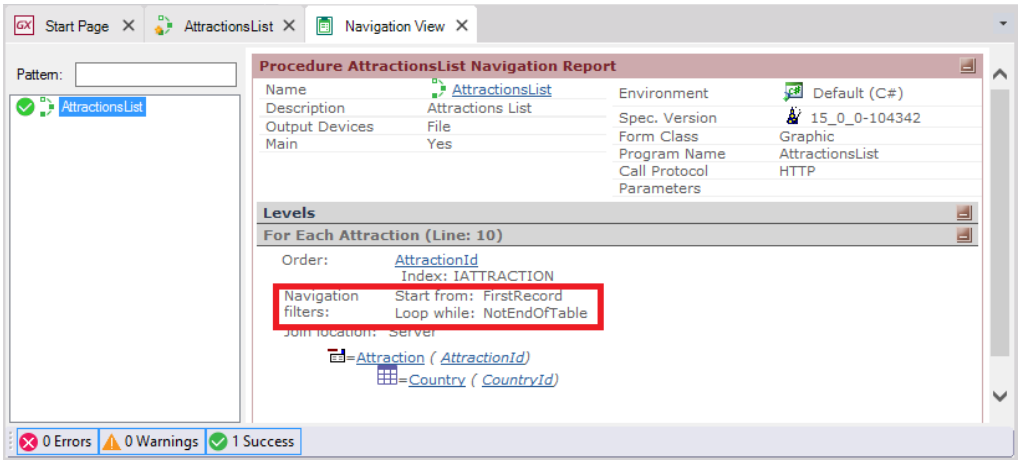


Também nos indica que, para ordenar o relatório de atrações

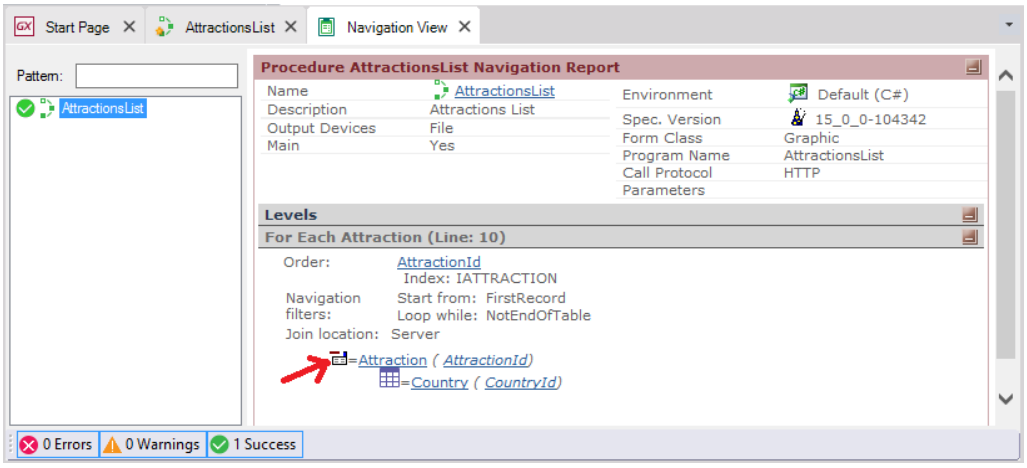


utilizou-se o atributo AttractionId (que é a chave primária da tabela Attraction)...

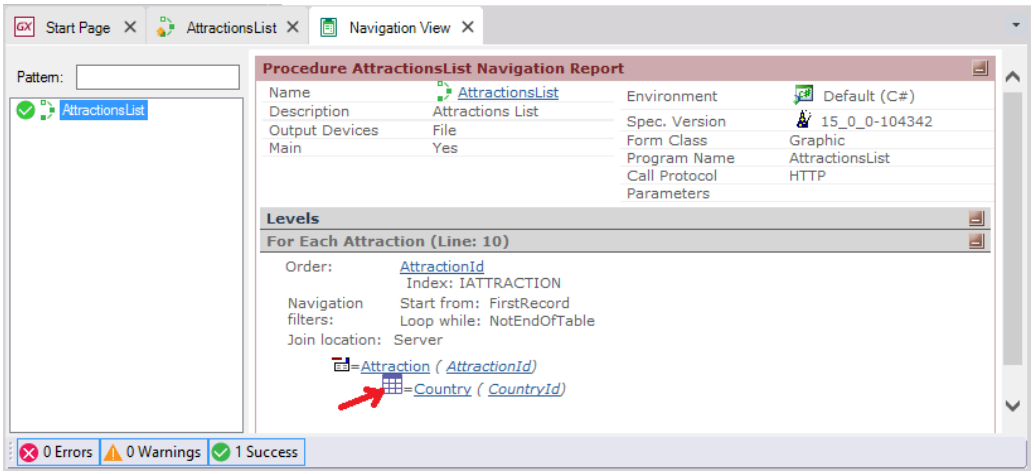
Nos mostra também que percorreu todos os registros da tabela: pois começou pelo primeiro registro e a percorreu até chegar ao fim da tabela.



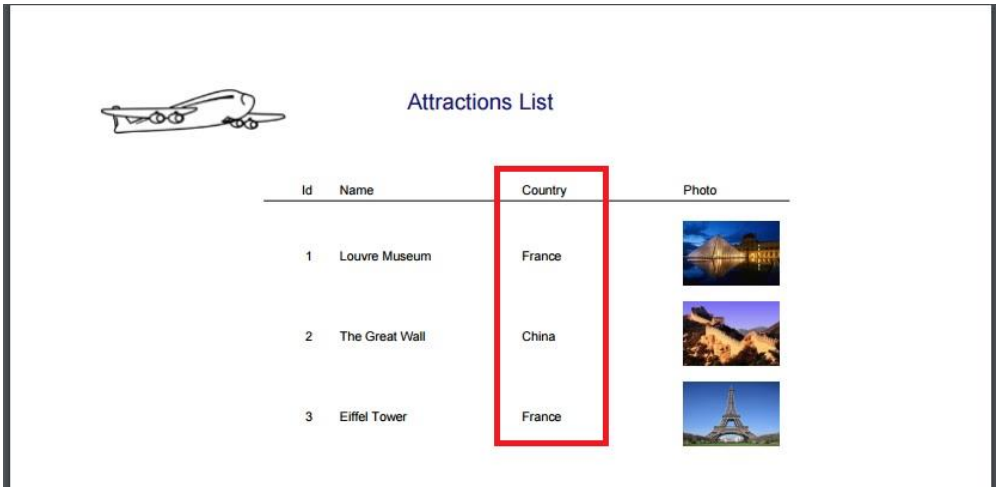
quer dizer que mostrou todas as atrações.... e por último... nos indica que a tabela que **navegou** foi Attraction



e que **deve acessar** também a tabela Country



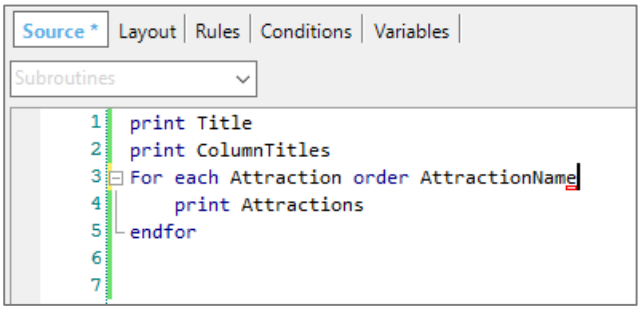
para recuperar determinada informação, já que no nosso relatório mostramos o nome do país.



Voltemos agora ao Source do procedimento

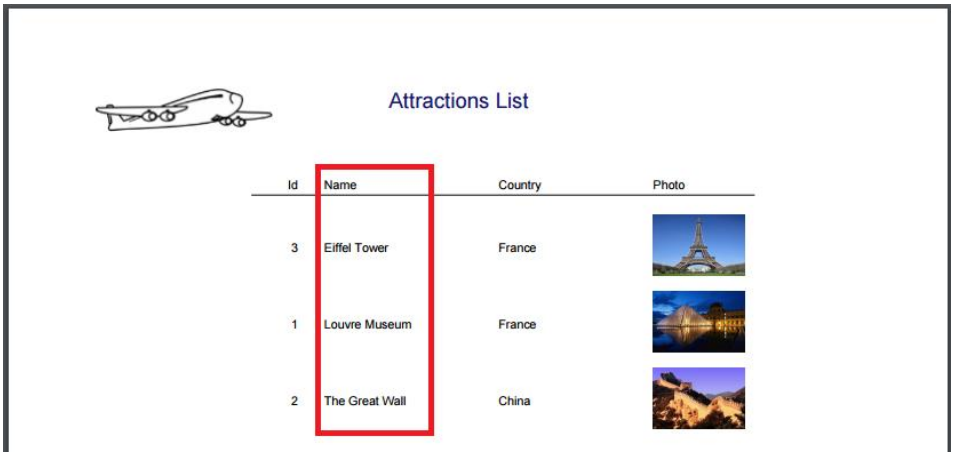
Uma coisa que havíamos deixado pendente é que as atrações seriam listadas em ordem alfabética, ou seja, por nome de atração.

E isto fazemos simplemente escrevendo logo depois do “For Each Attraction”, a cláusula “order AttractionName”

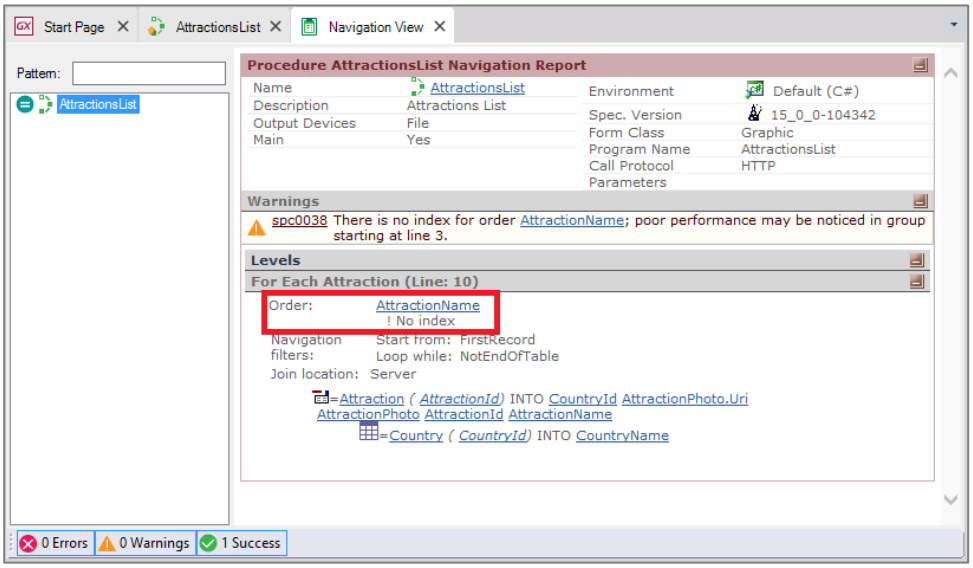


Vamos agora executar o procedimento para ver isso…

E vemos que agora as atrações turísticas estão ordenadas alfabeticamente.



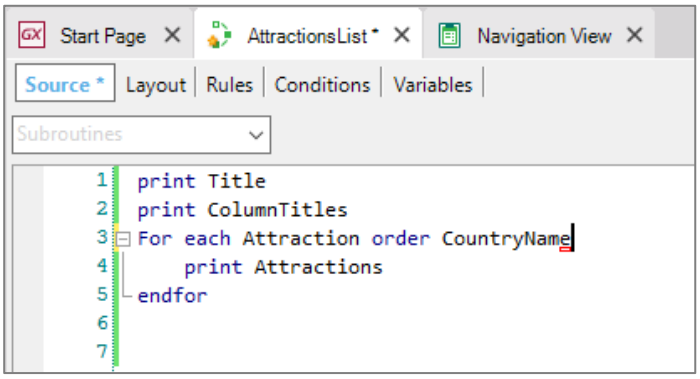
Se observarmos a lista de navegação, vemos que demonstra isso:



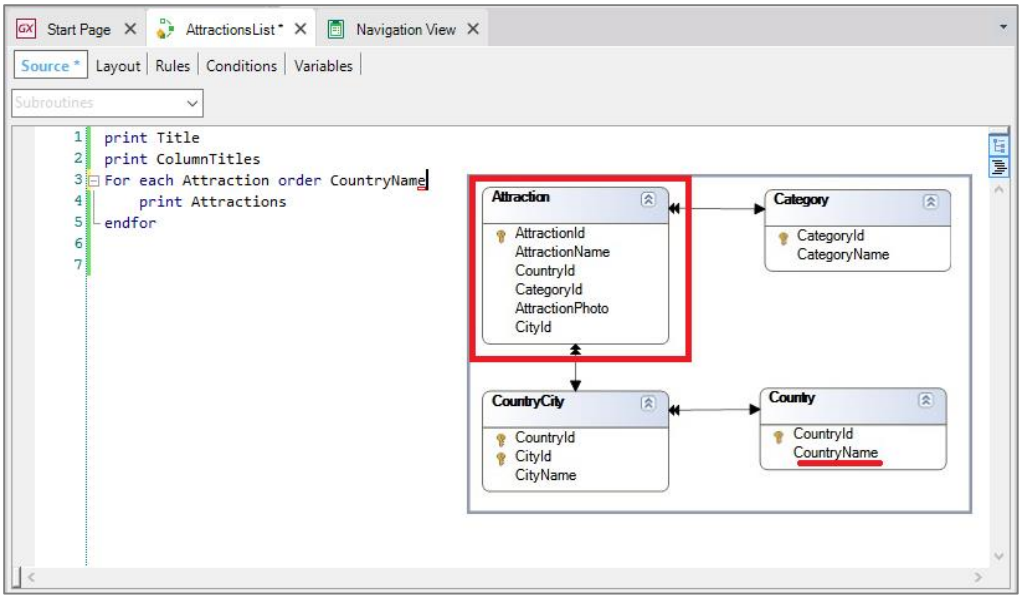
Por enquanto, não vamos tratrar a advertencia (Warning) que a lista de navegação está mostrando.

Agora, vamos observar algo interessante. É que GeneXus nos permite ordenar pelo valor de um atributo que não está na tabela ATTRACTION, mas sim em sua tabela extendida.

Vamos alterar o atributo do order para CountryName:



Este atributo não se encontra fisicamente na tabela base do For each



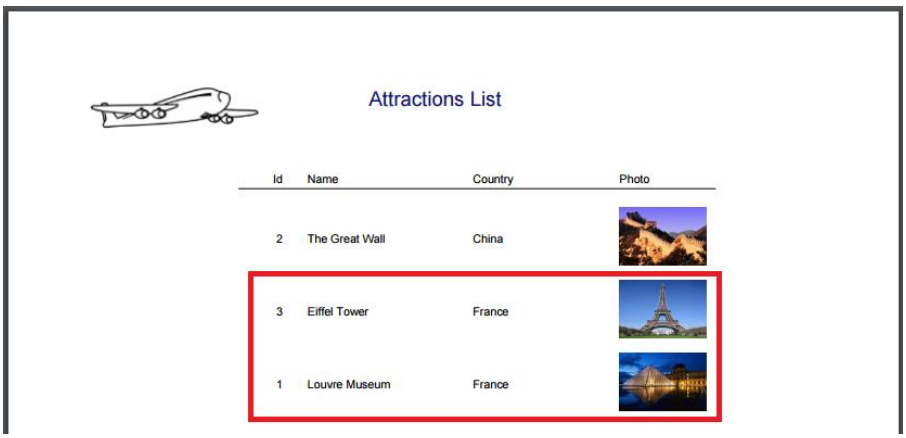
porém se encontra na tabela extendida da tabela base navegada, portanto podemos ordenar pelos atributos desta.

Executemos o procedimento para ver o resultado.

E vemos que agora saem listadas as atrações turísticas ordenadas pelo nome do país.

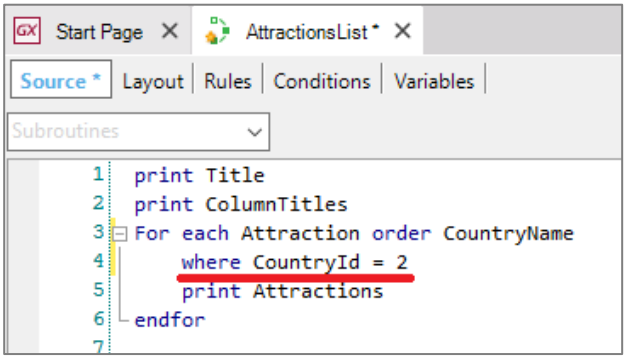
Assim como incluimos a **cláusula opcional "order"** ao For each, a sintaxe do For each permite que agreguemos várias cláusulas e definições opcionais, como veremos.

Por exemplo, o que aconteceria se na agência de viagens nos pedem que listemos somente as atrações turísticas da França?



Somente adicionaremos ao comando For Each, uma cláusula chamada **Where**, para que filtre e mostre unicamente os dados que cumpram com a condição desejada.

Então, depois do comando For Each adicionamos uma nova linha e escrevemos Where…CountryId=2, já que sabemos que o Id da França é o 2.



Pressionamos F5.

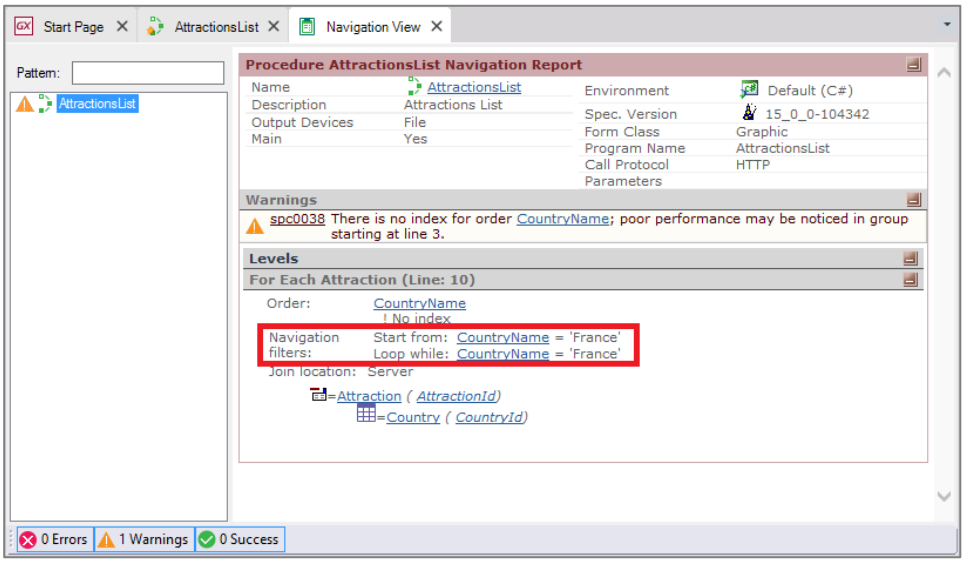
No lugar de filtrar pelo identificador do país, podíamos ter escrito **Where CountryName='France'**.

```
1  print Title
2  print ColumnTitles
3  For each Attraction order CountryName
4       where CountryName = 'France'
5       print Attractions
6  endfor
```

Salvamos e executamos outra vez o relatório….

E vemos que o resultado é exatamente o mesmo.

Se observarmos agora a lista de navegação:

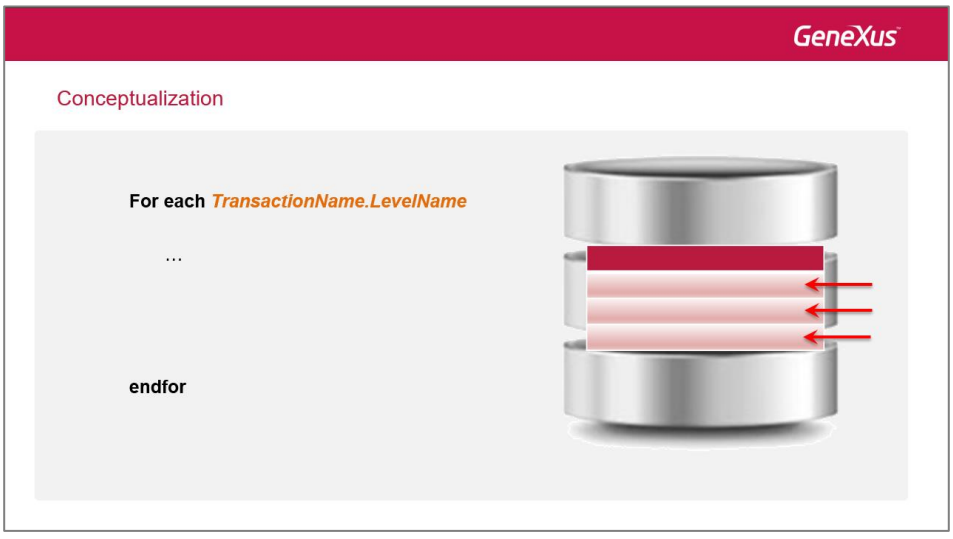


Vemos que já não se percorre toda a tabela Attractions. Como estamos ordenando por CountryName, para nos mostrar os países de nome 'France', tem que percorrer somente uma parte menor da tabela e não toda. É como buscar num dicionário a palavra 'France'. Não buscamos em todo o dicionário. Vamos diretamente ao "F".

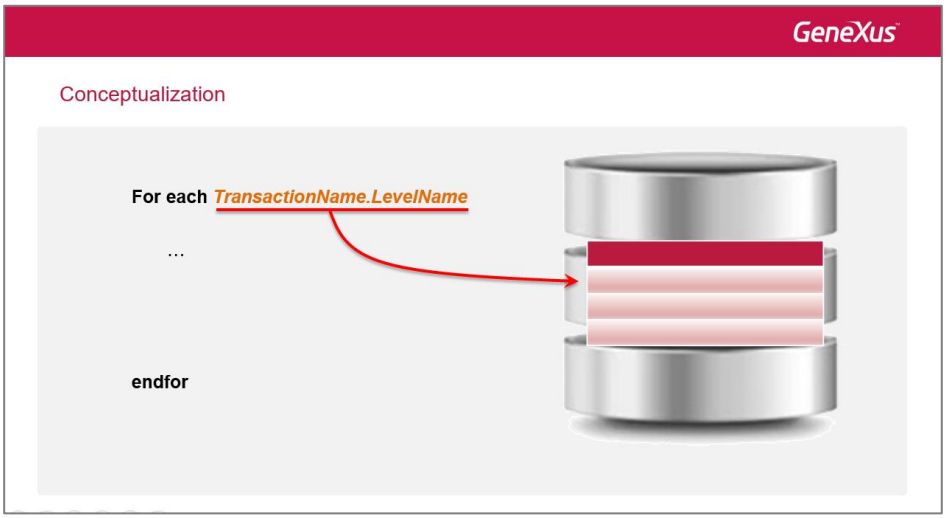
Finalmente atualizamos as mudanças no GeneXus Server.

Repassemos agora os conceitos aprendidos:

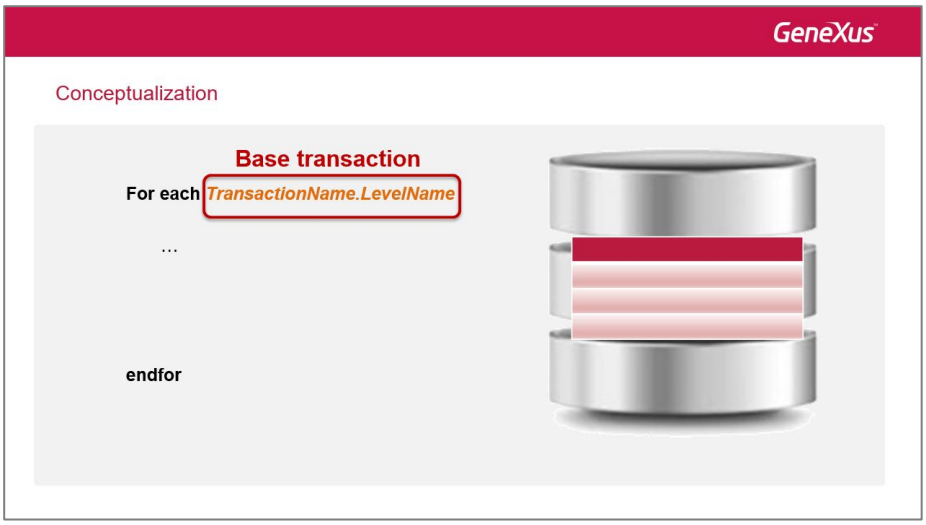
O comando For each é utilizado para percorrer cada registro de uma tabela e fazer algo com a informação relacionada.



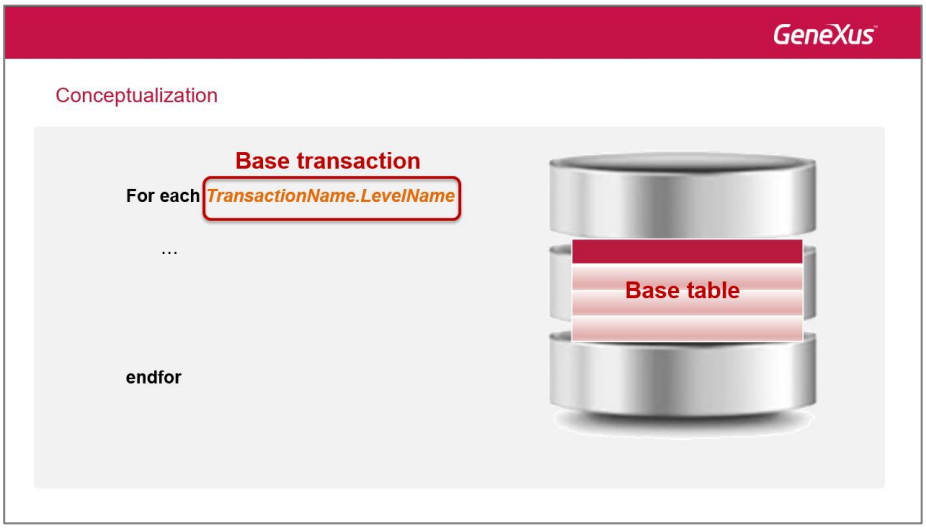
Para isso, indicamos o nome do nível da transação cuja tabela associada queremos percorrer.



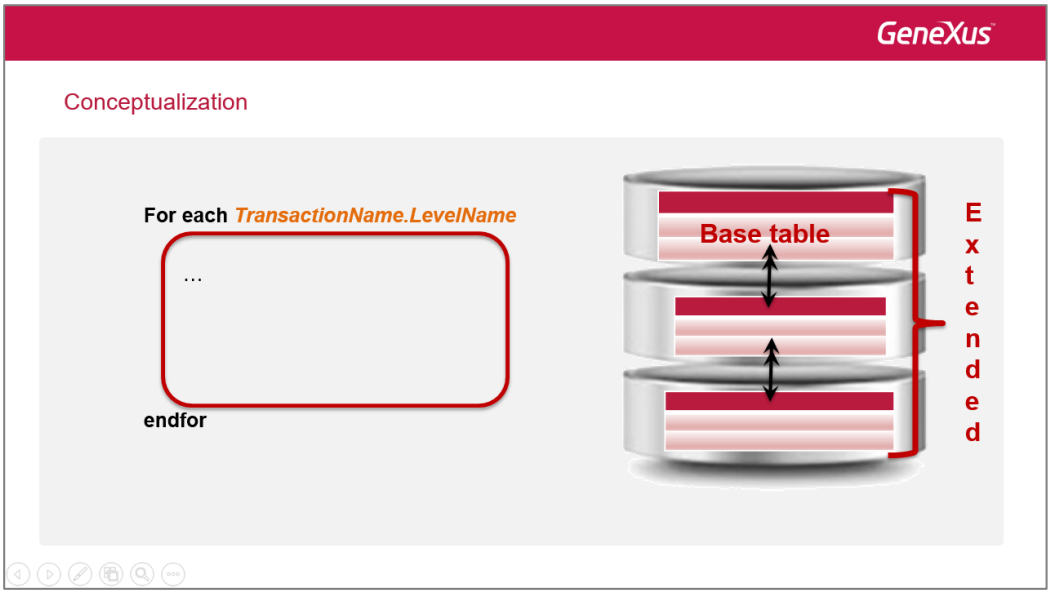
A esta indicação de nível chamamos **transação base** do for each:



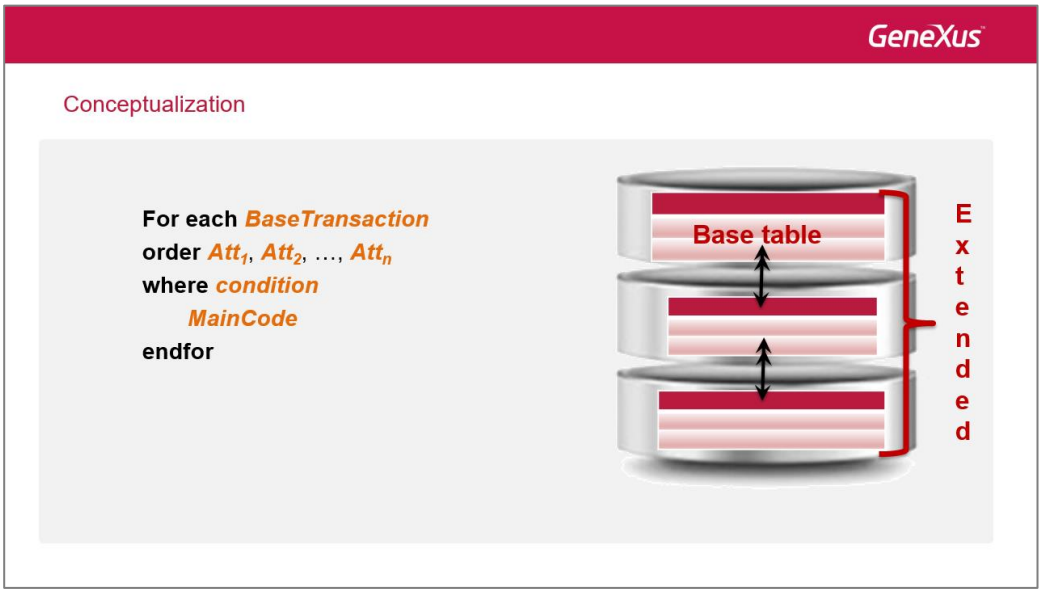
E desse nível, GeneXus determinará a tabela a percorrer, o que chamamos **tabela base** do for each.



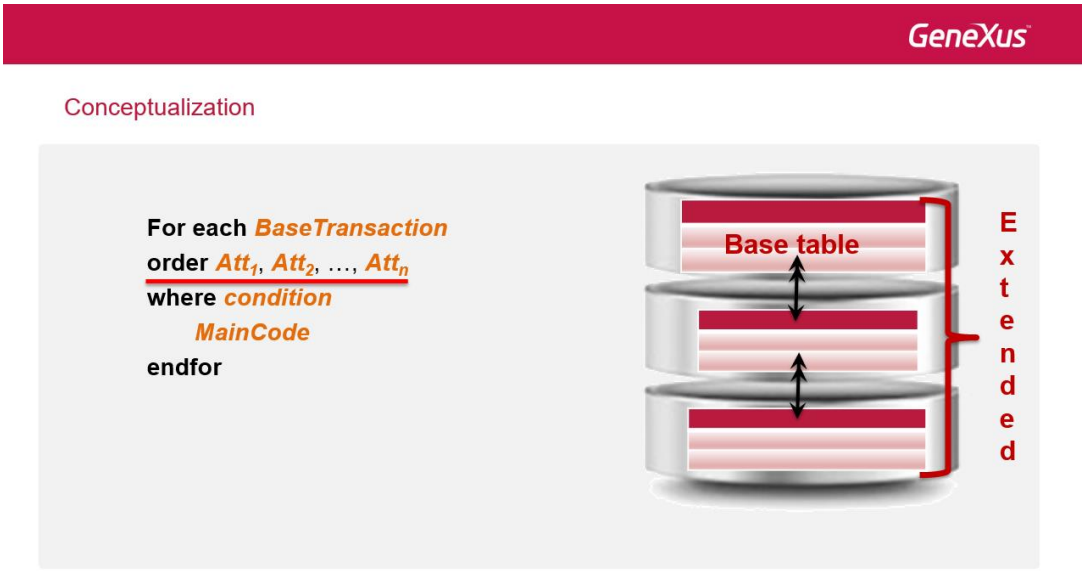
O conjunto de atributos que estão entre o For each e o Endfor devem pertencer à tabela extendida dessa tabela base.



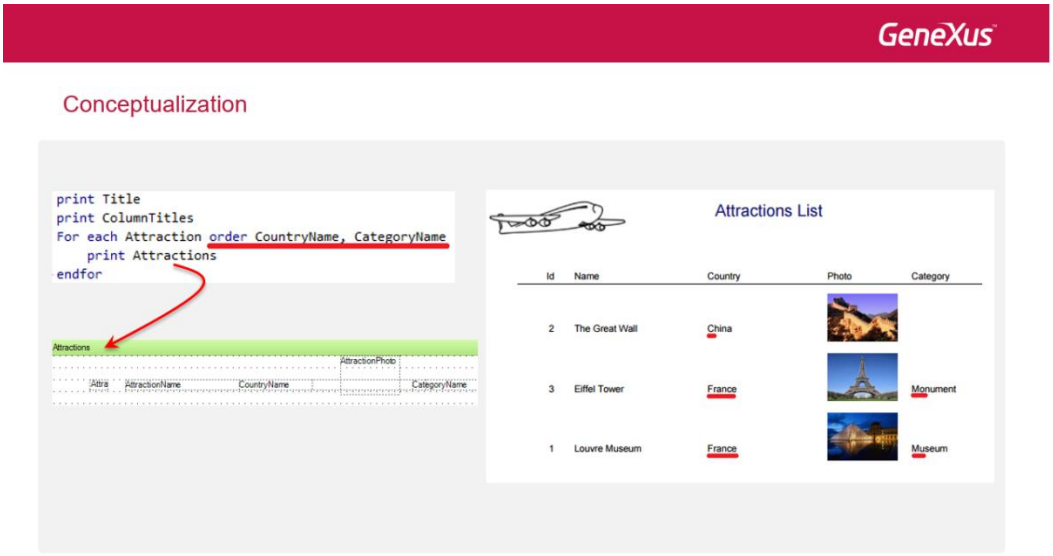
Até aqui, resumimos o que vimos até o momento sobre o comando for each:



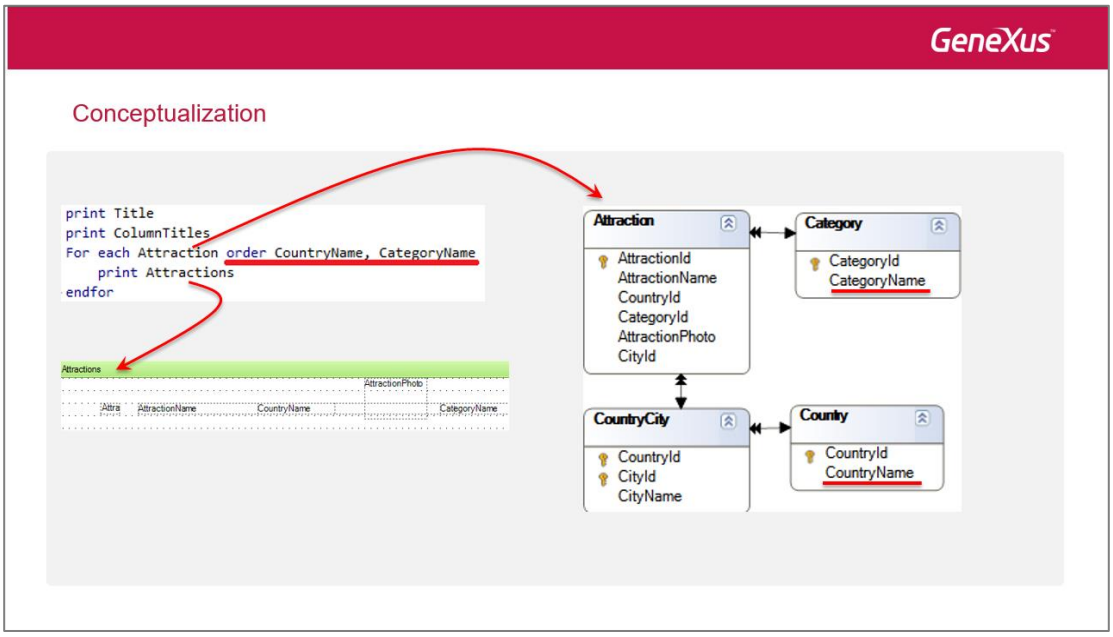
Através da cláusula **Order** é possível indicar o critério pelo qual iremos ordenar a informação devolvida pelo For each. Esta ordenação pode ser baseada em atributos da tabela base do For each ou de sua tabela extendida. Veremos ainda que podemos ordenar por só um atributo, ou por vários.



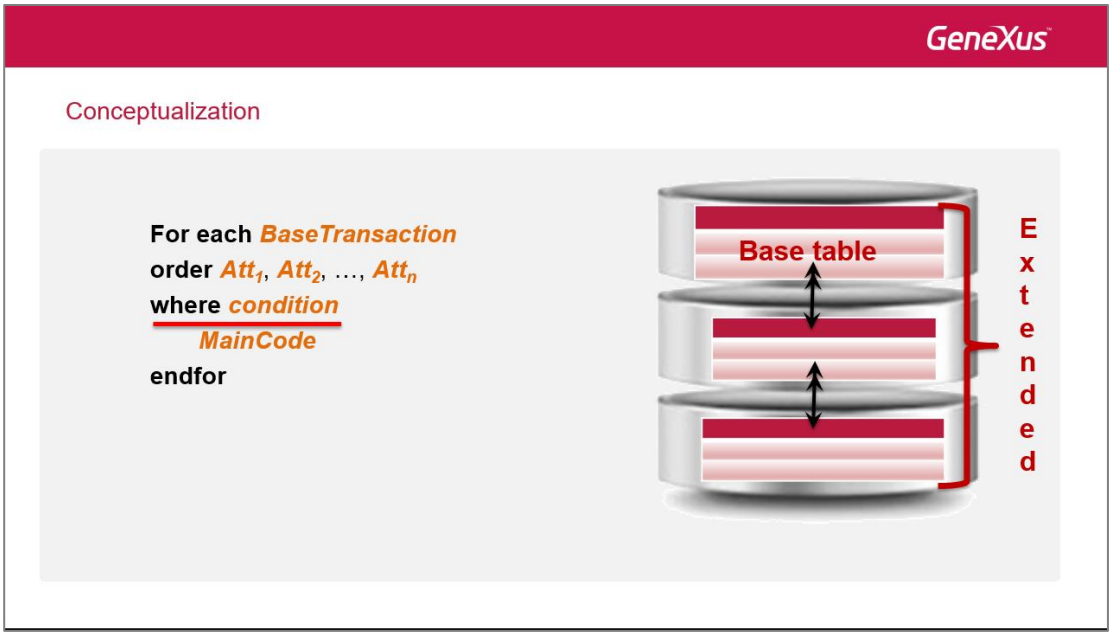
Por exemplo, se mostrássemos também a categoria da atração turística no relatório e quiséssemos ordenar pelo nome do país e, dentro das atrações que são do mesmo país, pelo nome de categoría… escreveríamos ambos atributos em sequência: primero CountryName e em seguida CategoryName…



Aqui, tanto CountryName, como CategoryName não estão presentes na tabela base Attraction, mas sim em sua tabela extendida:

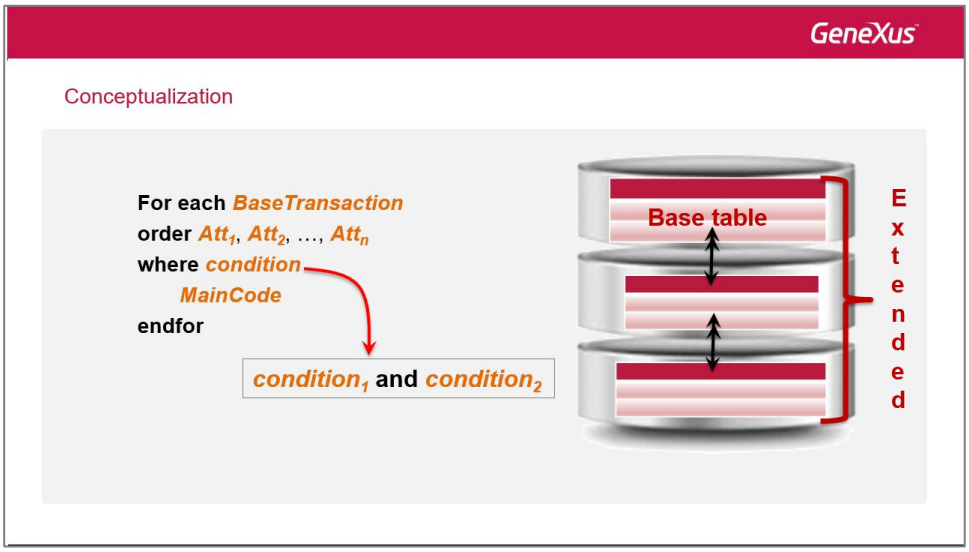


Para filtrar a información devolvida pelo For each, se utiliza a cláusula **Where**, onde especificamos a condição que deverão cumprir os registros para serem filtrados:

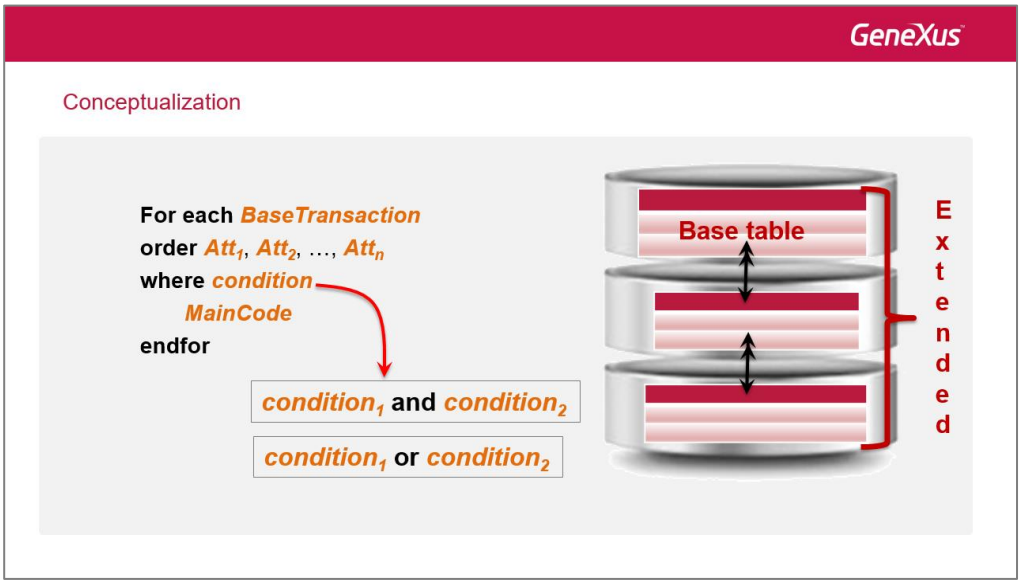


A condição pode ser complexa, incluindo várias condições unidas por AND ou OR. Por exemplo:

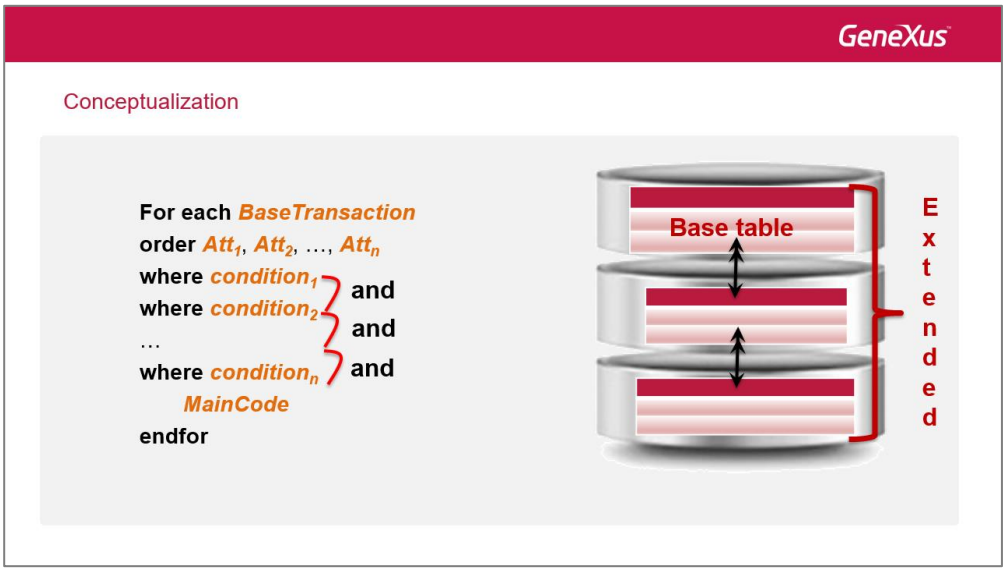
- Condition1 **AND** Condition2: que significa que ambas devem ser verdadeiras para que o registro seja filtrado.



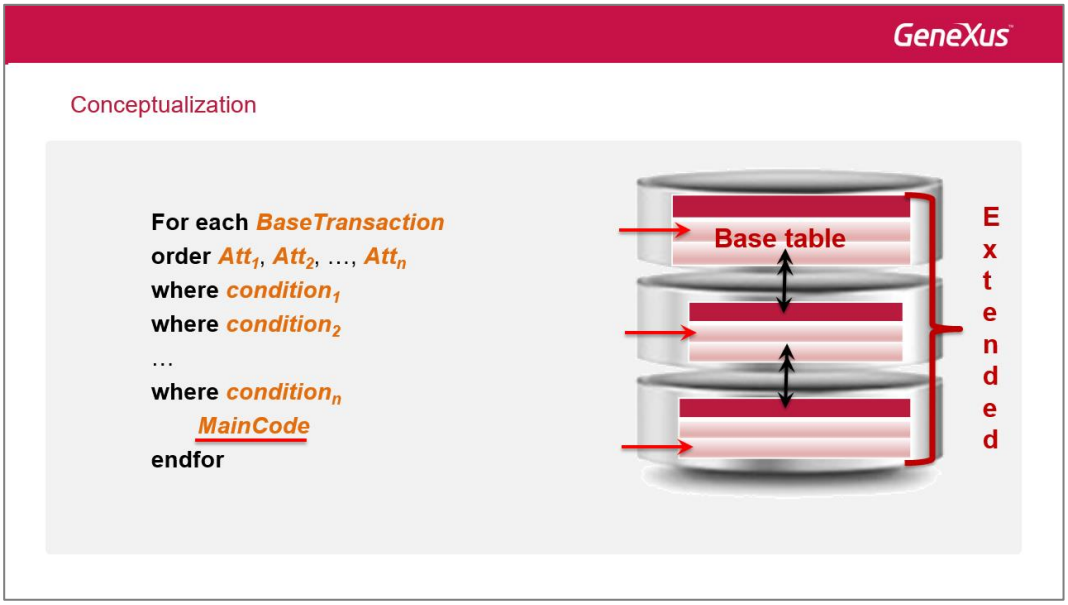
- Condition1 **OR** Condition2: que significa que se uma delas for verdadeira, já é suficiente para que o registro seja filtrado:



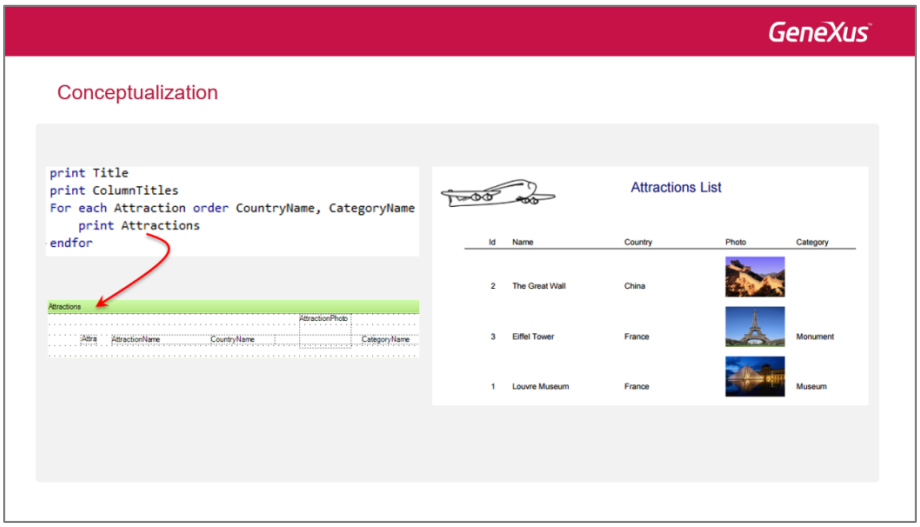
Também podemos colocar várias cláusulas **Where**, o que é o mesmo que escrever somente uma, com as condições unidas por **AND:**



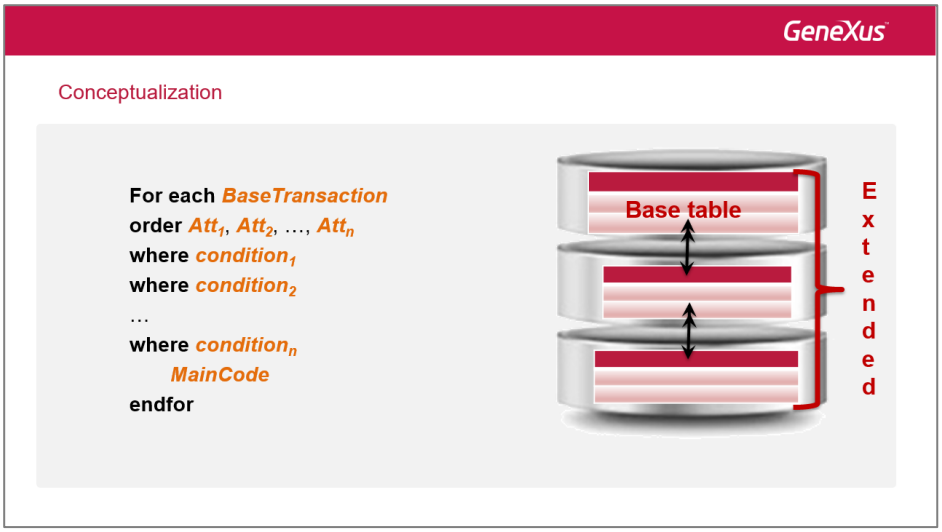
Dentro do comando for each, em seu **código principal**, se escrevem os comandos que queremos que se executem, um após outro, para realizar passo a passo o que se deseja fazer com as informaçãoes do registro da tabela base que se está navegando no momento… e os associados por tabela extendida:



Por exemplo, imprimir un printblock:



Assim temos, portanto, a estrutura do comando for each:



O comando admite mais cláusulas e opções. Algumas serão vistas em outros vídeos. Outras serão abordadas em outros cursos.