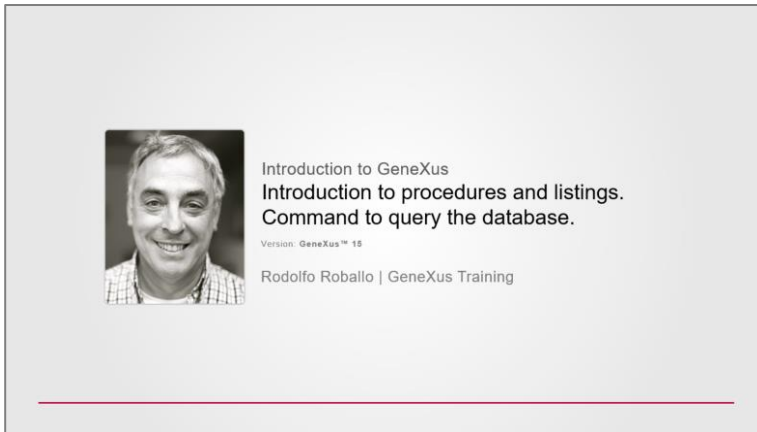


Introducción a procedimientos y listados. Comando para consultar la base de datos.



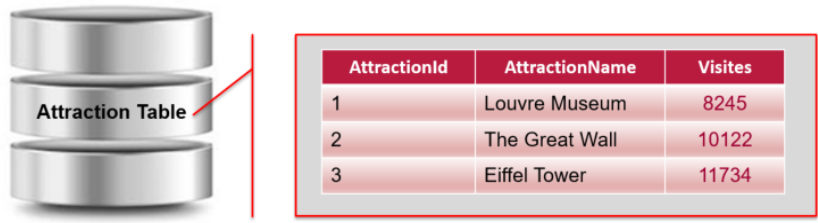
Pasemos a conocer los objetos de tipo **Procedimiento** que podemos definir en una base de conocimiento GeneXus



Este tipo de objeto nos permite definir **PROCESOS para acceder y navegar tablas de la base de datos con distintos objetivos:**

→ por ejemplo, podríamos necesitar navegar los registros de cierta tabla, que cumplan con determinadas condiciones y actualizar para dichos registros cierto atributo con determinado valor

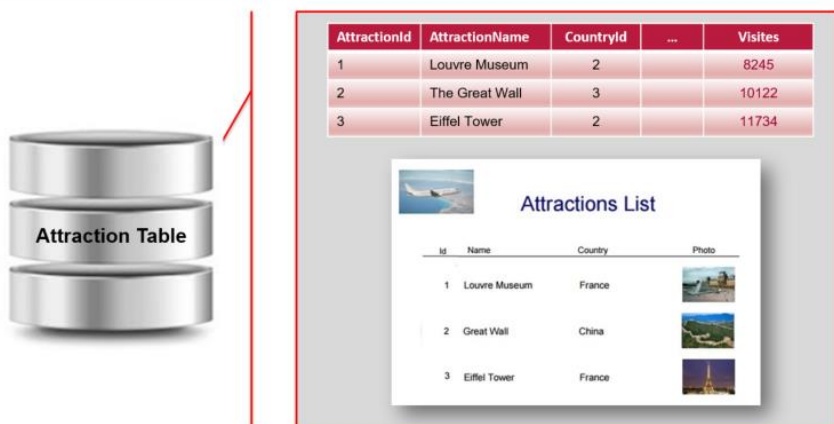
GeneXus



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




→ o navegar cierta tabla, e imprimir todos sus datos en un listado PDF, ordenados por algún criterio

GeneXus



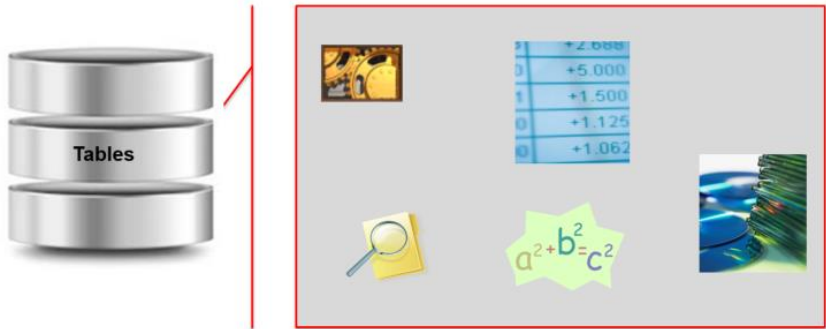
AttractionId	AttractionName	CountryId	...	Visites
1	Louvre Museum	2		8245
2	The Great Wall	3		10122
3	Eiffel Tower	2		11734


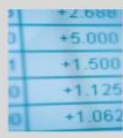
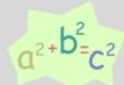

Attractions List

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

→ o también definir procesos específicos que necesitamos resolver, que contengan búsquedas, cálculos, actualizaciones a la base de datos y que vayan imprimiendo información.

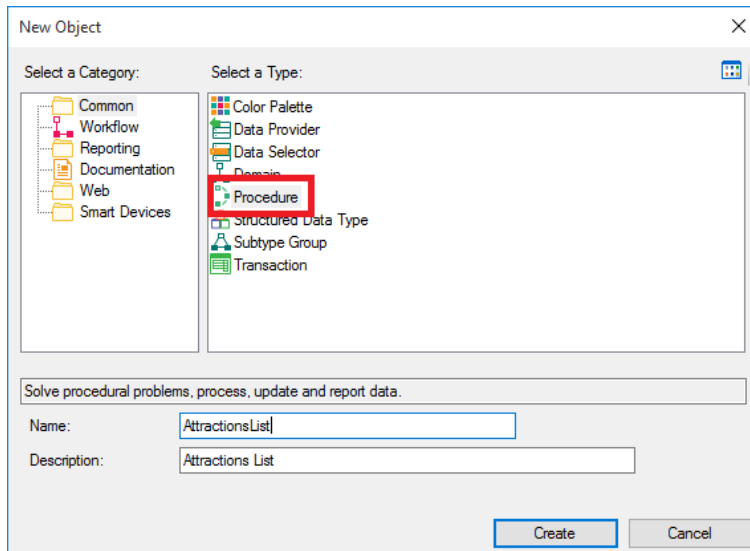
GeneXus



En primer lugar comencemos por definir un procedimiento para listar todas las atracciones turísticas que ofrece la agencia de viajes, ordenadas alfabéticamente.

Así que creamos un nuevo objeto, esta vez de tipo procedure, al que llamamos:

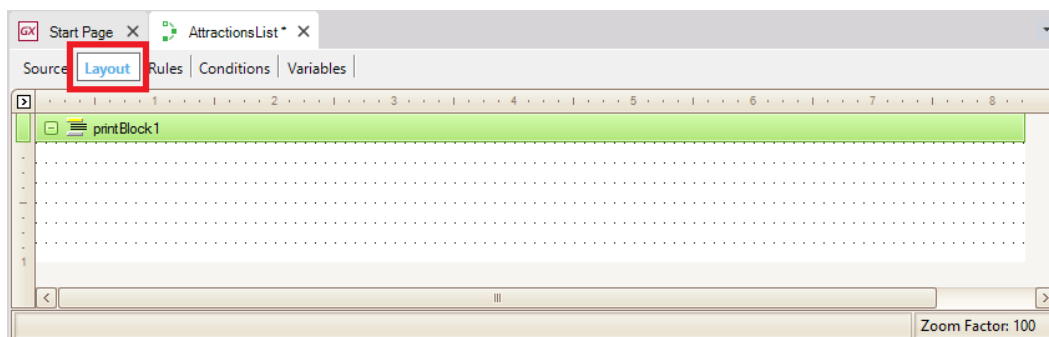


Una vez creado el objeto vemos que GeneXus nos posiciona en una sección llamada Source:



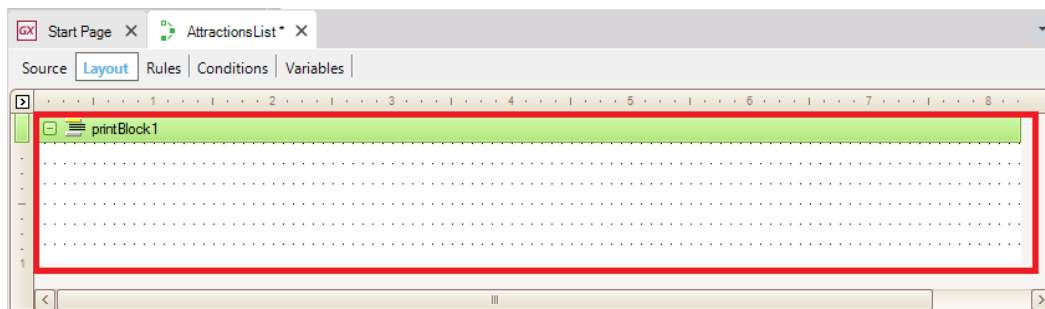
Aquí es donde escribiremos comandos y órdenes que permitan al procedimiento cumplir con el objetivo para el cual lo hemos creado, en nuestro caso imprimir un listado de atracciones turísticas.

Ahora observemos esta otra sección llamada Layout:



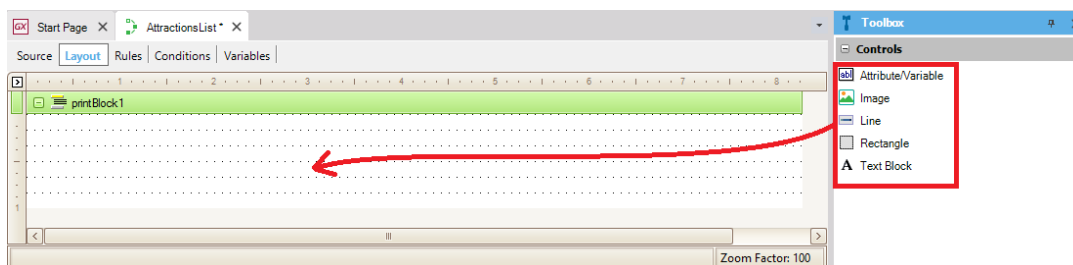
El layout es el lugar destinado al diseño de la salida, es decir, donde especificaremos cómo deseamos ver nuestros datos.

Se compone de **printblocks**



y dentro de los printblocks incluiremos lo que deseamos mostrar.

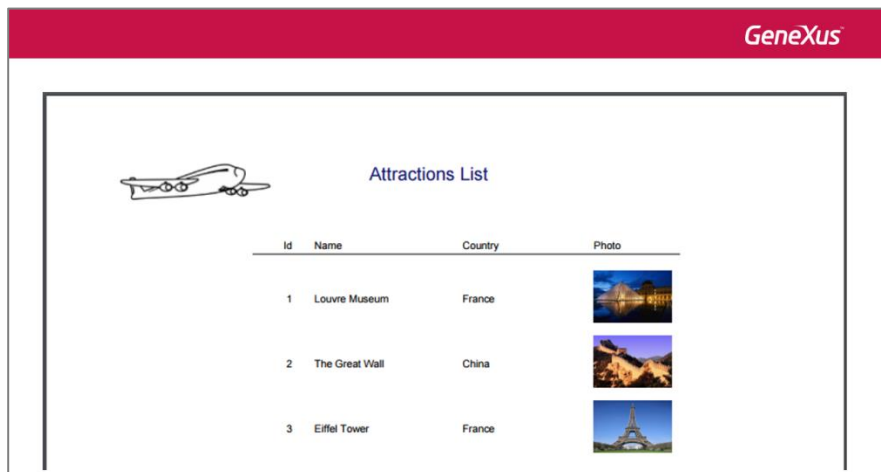
Podemos querer mostrar títulos, líneas, rectángulos, imágenes, así como también valores de atributos o variables. Para ello los arrastraremos dentro del printblock.



Observemos que el layout automáticamente contiene un printblock

En este printblock, podemos querer incluir un título, o la fecha del día, y también podremos agregar más printblocks en esta sección, como veremos.

Pensemos ahora **cómo queremos que se vea nuestro listado**, así podremos proceder a definirlo. Podría ser algo así...



con una imagen un título

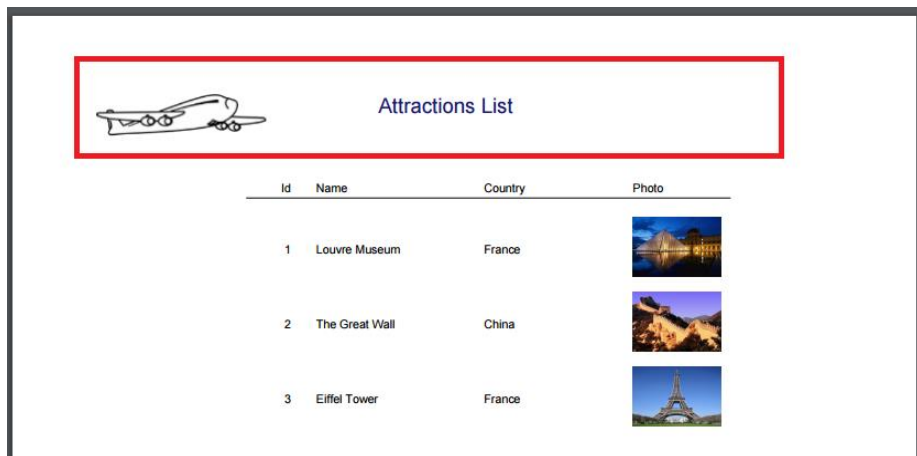
columnas

y todas las atracciones turísticas ordenadas alfabéticamente

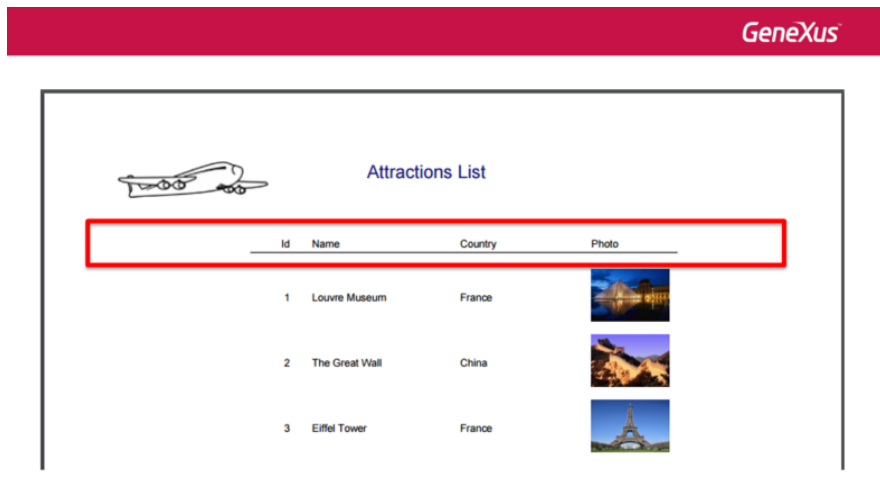
Notemos que queremos mostrar 2 contenidos fijos, como es el título del listado con la imagen y los títulos de las columnas con la línea debajo

y luego, la información que queremos mostrar de las atracciones la cual irá cambiando línea a línea, ya que queremos mostrar los datos que tengamos almacenados en nuestra base de datos.

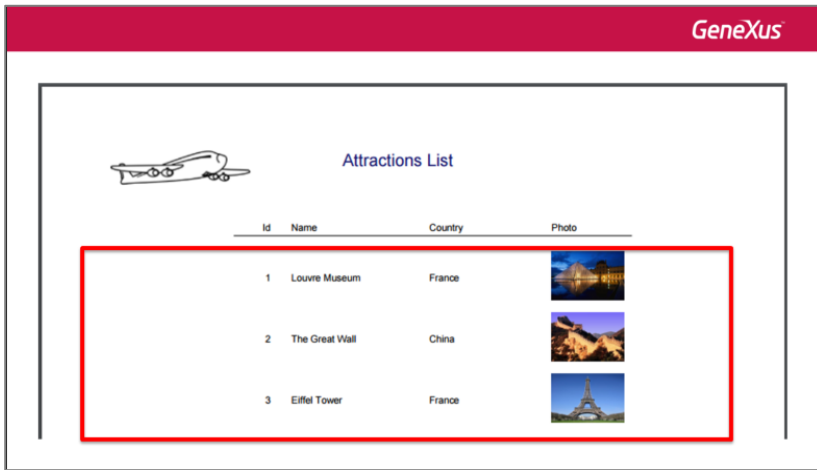
Podríamos definir entonces tres printblocks: uno en el cual incluiríamos el título del listado y la imagen



otro para mostrar los títulos de las columnas con la línea debajo



y un tercer printblock

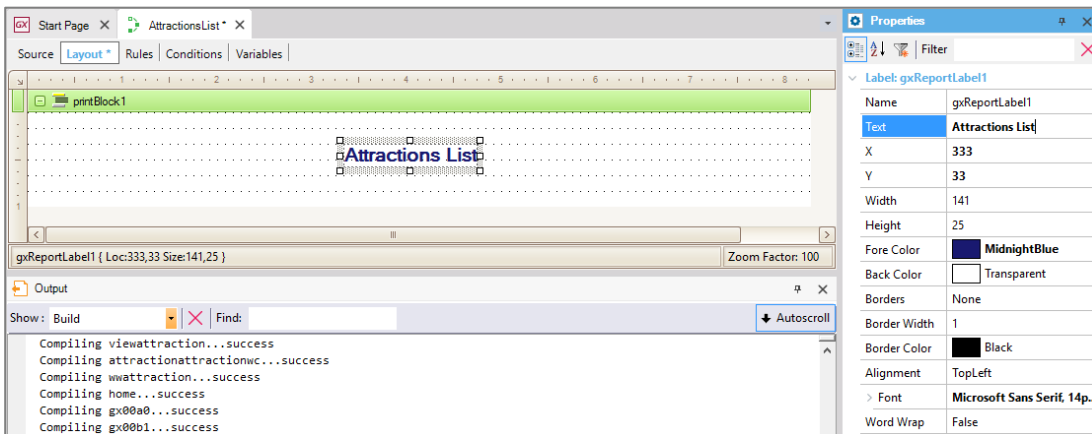


en el cual vamos a desplegar los datos de las atracciones turísticas.

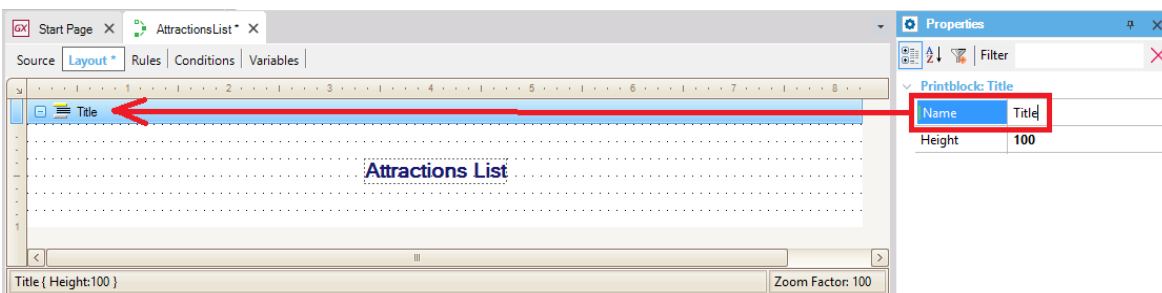
Empecemos entonces a definir esto.

Podemos usar el printblock que se creó cuando creamos el objeto procedimiento para el título y la imagen.

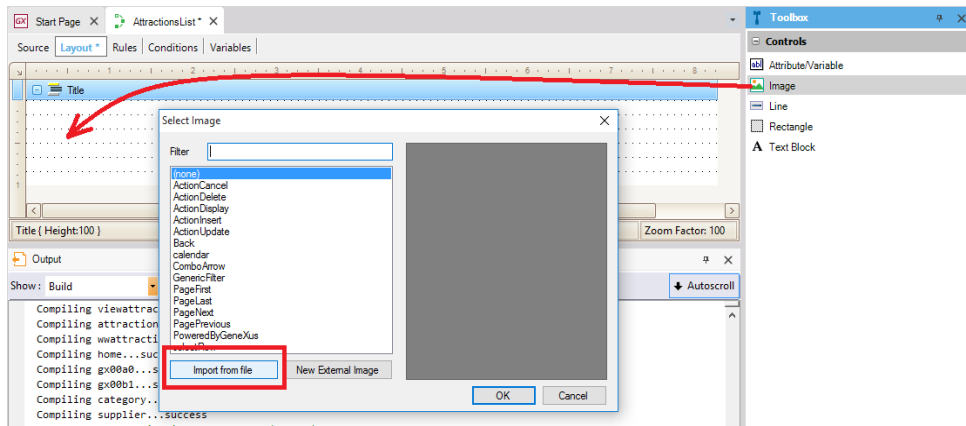
Empecemos por el título. Para esto desde la Toolbox... arrastramos el control Text Block... editamos sus propiedades... y en la de nombre Text escribimos "AttractionsList". Modificamos también su color, su fuente..., y lo ubicamos donde queremos que aparezca en relación a los márgenes.



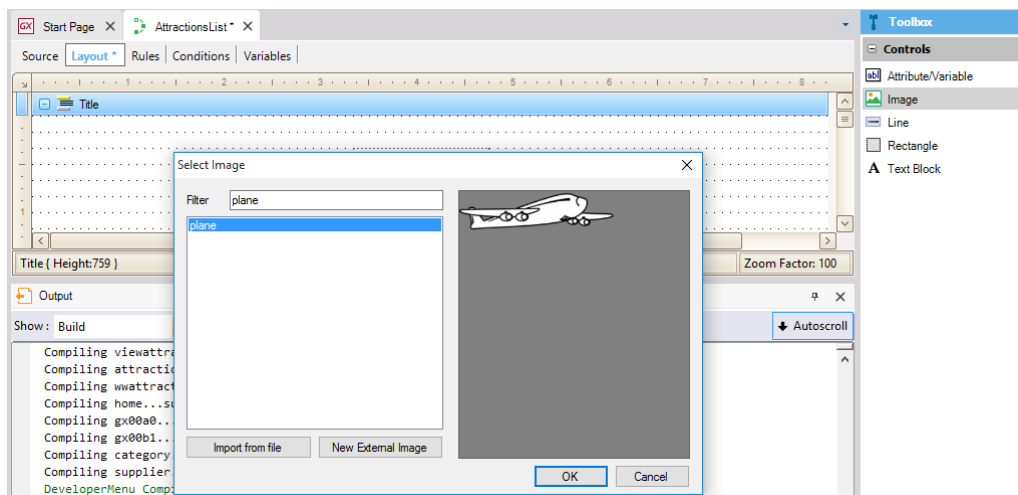
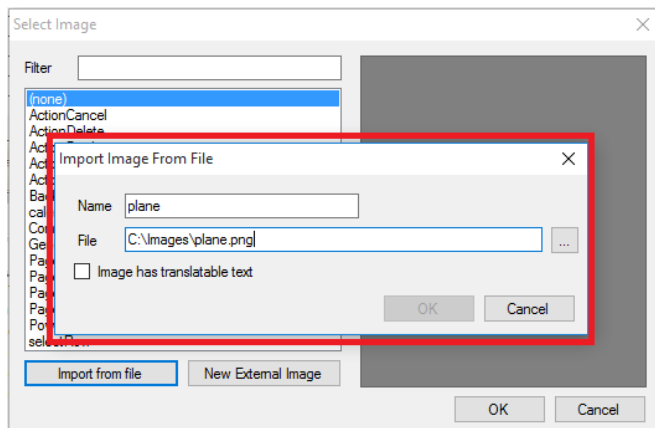
Nombraremos a este printblock con un nombre claro que represente lo que está mostrando. Así que seleccionamos las propiedades del printblock y editamos su propiedad Name, asignándole como nombre "Title".

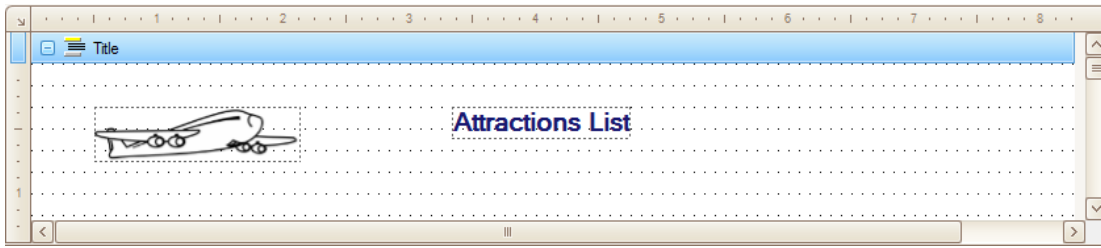


Insertemos, ahora, la imagen con el avión, a la izquierda. Para ello arrastramos desde la Toolbox el control Image y lo soltamos donde deseamos colocarla. Observemos que se nos abre esta ventana que nos permite seleccionar alguna de las imágenes existentes en la base de conocimiento, o incorporar una nueva, por ejemplo importándola de un archivo:

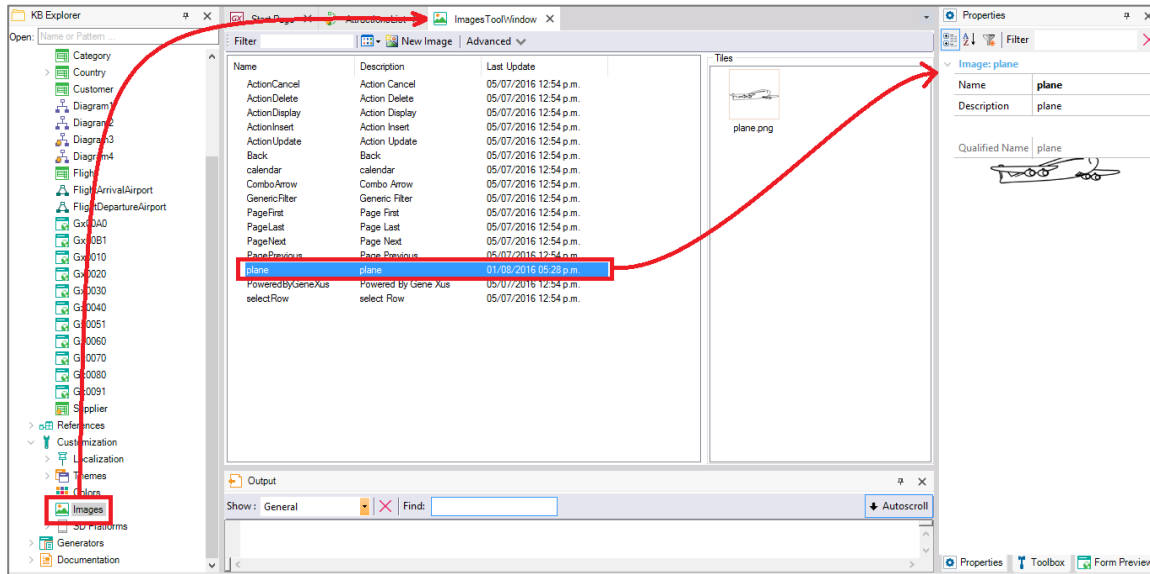


Este botón nos permite explorar nuestro sistema de archivos y elegir la imagen, que se creará como objeto GeneXus de tipo Image con el mismo nombre del archivo imagen como nombre por defecto. De allí en más podremos utilizar la imagen dentro de nuestra KB libremente.

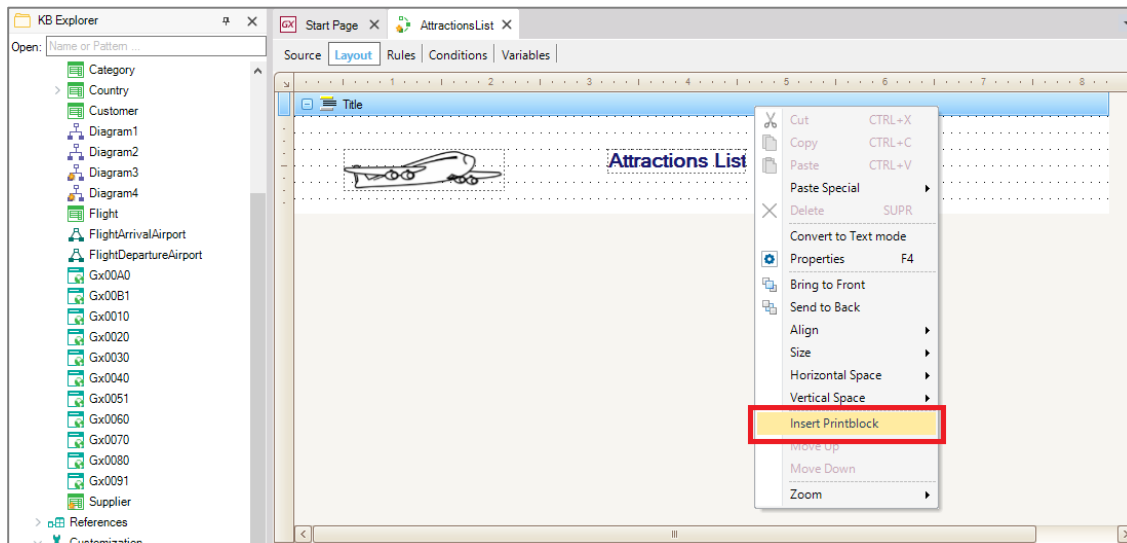




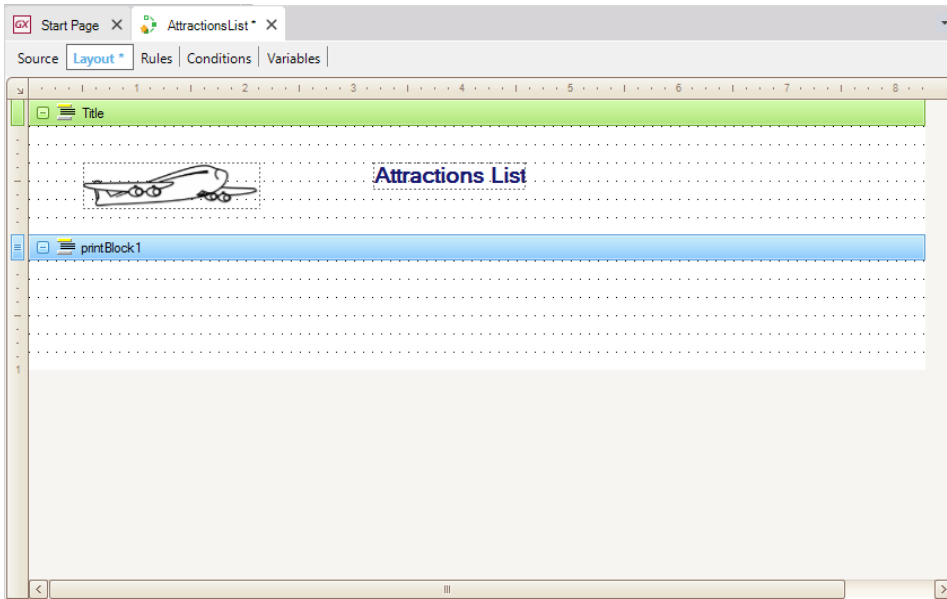
Observemos que aquí podemos acceder a todas las imágenes de la KB entre las que se encuentra la de nuestro avión.



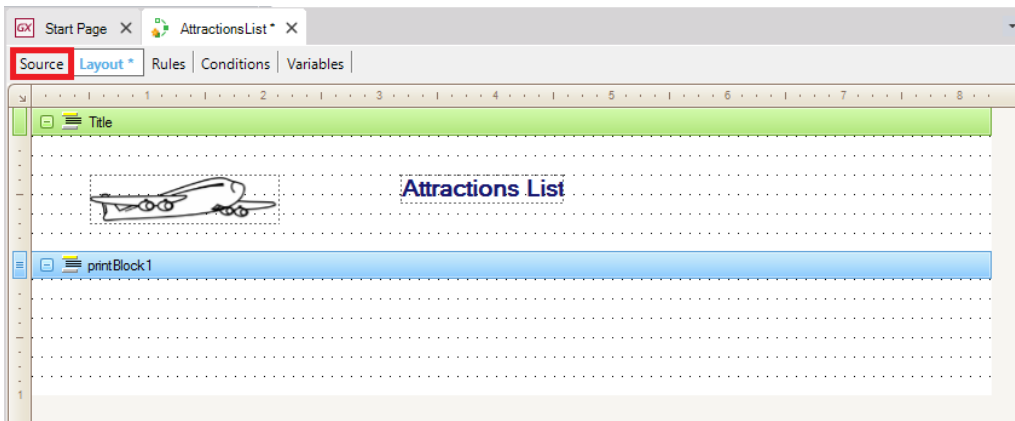
Ahora vamos a crear otro printblock para incluir en él los títulos de las columnas, con una línea debajo. Si presionamos el botón derecho del mouse **sobre determinado printblock** y seleccionamos la opción “Insert Printblock”



se insertará un nuevo printblock **debajo de él**.

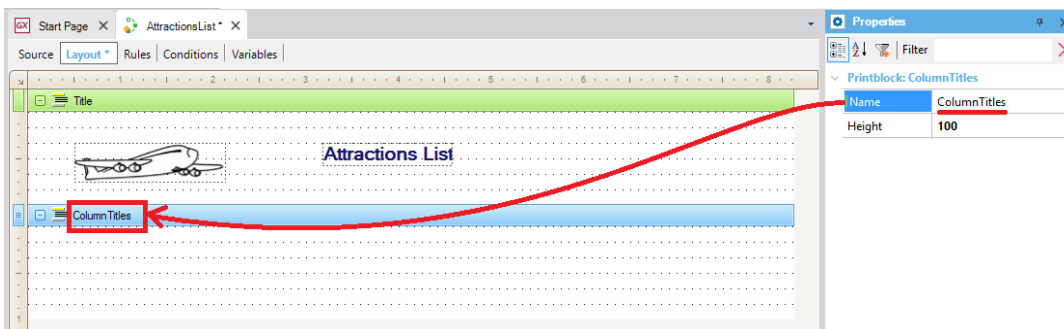


El orden de los printblocks en el Layout no es importante, ya que no será necesariamente el orden en el que se imprimirán. **Definimos cuándo imprimir cada printblock** en el código que vamos a escribir en el Source del procedimiento.



Veremos esto en breve.

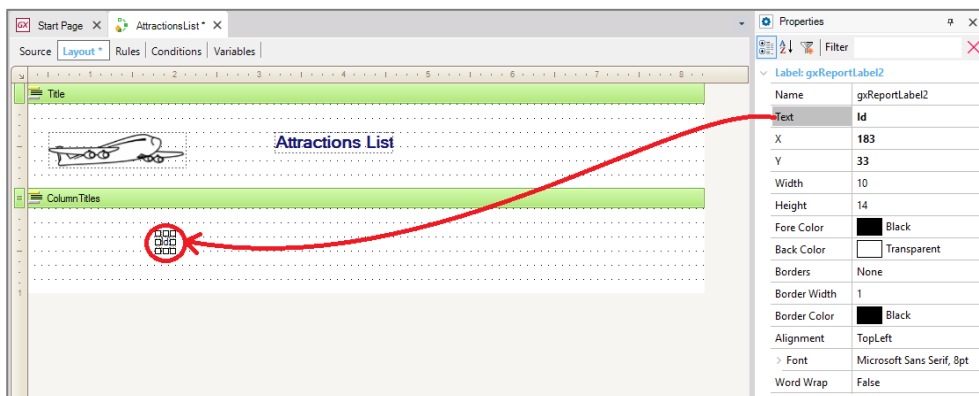
Ahora a este nuevo printblock le pondremos como nombre “ColumnTitles”



... y ahora vamos a insertar en este nuevo printblock un TextBlock por cada texto que queremos mostrar como título de columna.

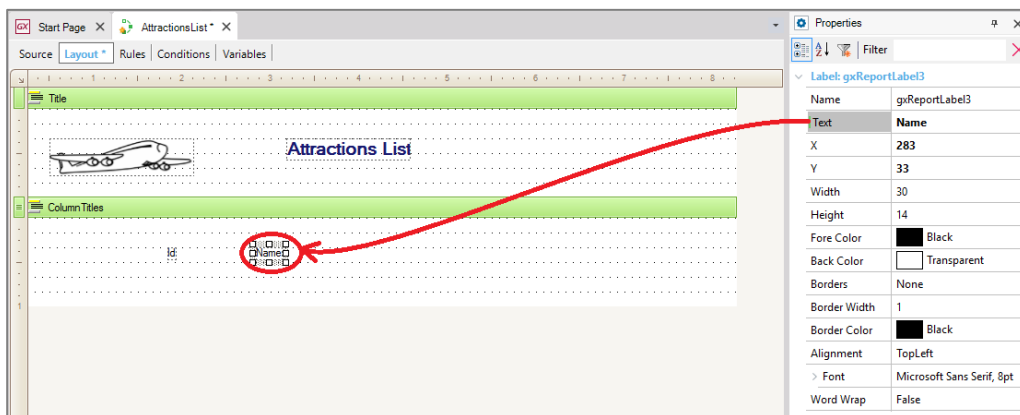
Así que desde la Toolbox, arrastramos el text block

Y en su propiedad Text escribimos "Id".



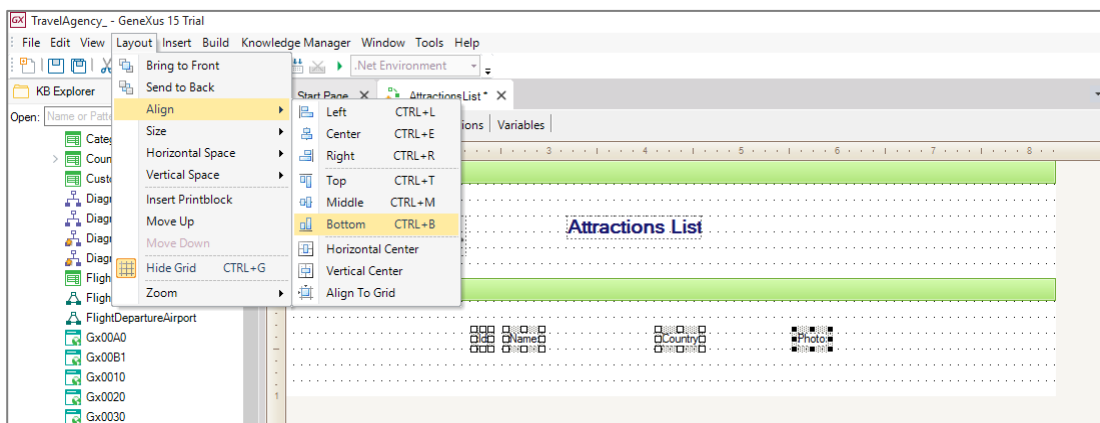
Agregamos otro Textblock:

Y en su propiedad Text agregamos el texto "Name":



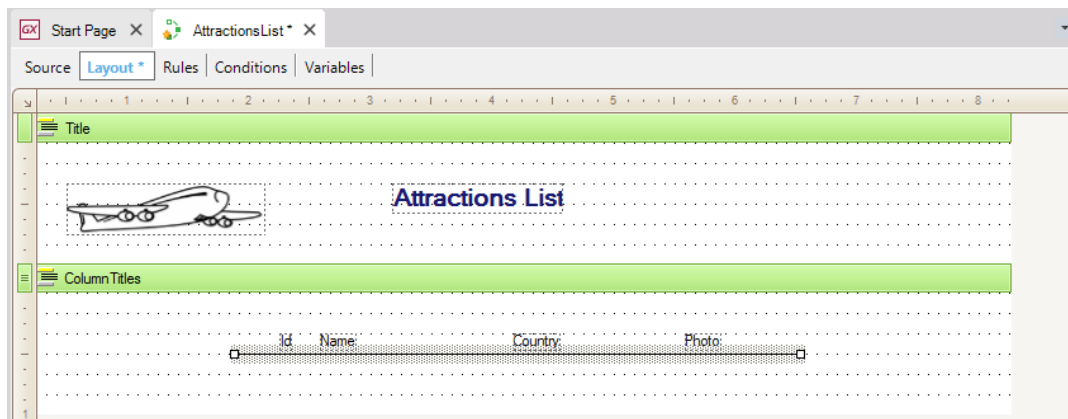
y ahora otro Textblock para mostrar el texto "Country". Y por último un text block para el título "Photo".

Ubicamos los controles en la posición que queremos... Los podemos alinear seleccionándolos todos y luego:

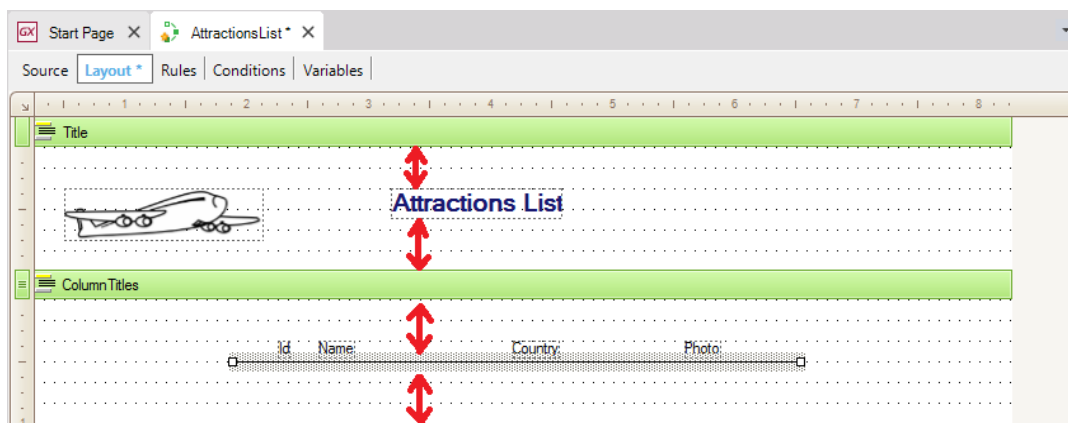


Por último, vamos a insertar una línea debajo de estos títulos de columnas.

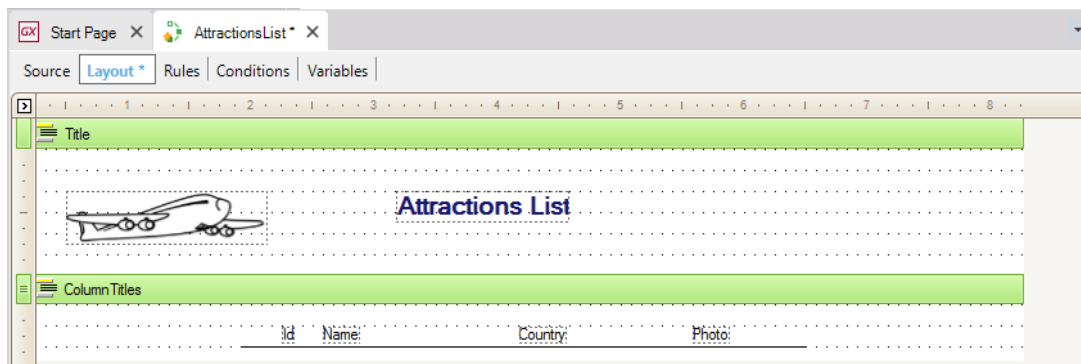
Así que volvemos a la ToolBox, y arrastramos un control "Line".
Arrastramos desde aquí... dando el largo que deseamos...



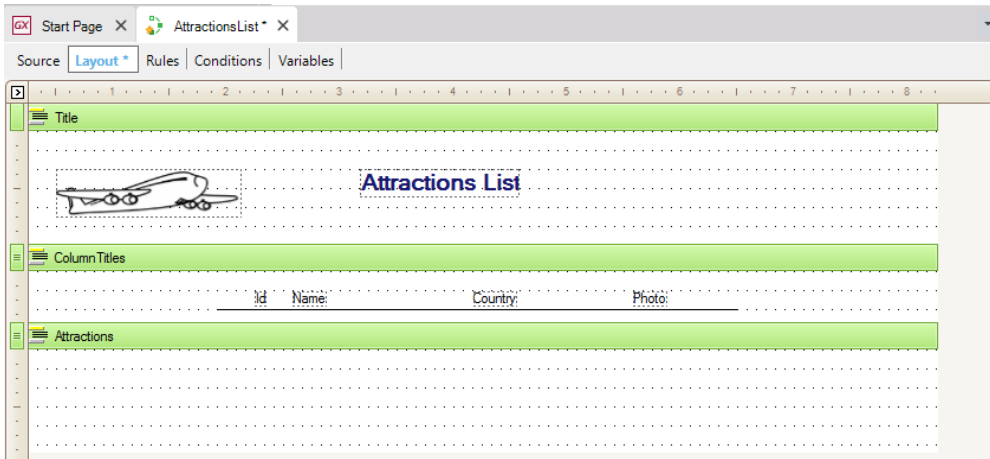
Si ahora en el Source escribiéramos las órdenes para que se imprima el primer printblock y a continuación el segundo, estos espacios en blanco de aquí serán respetados:



Vamos a reducir los del segundo printblock:

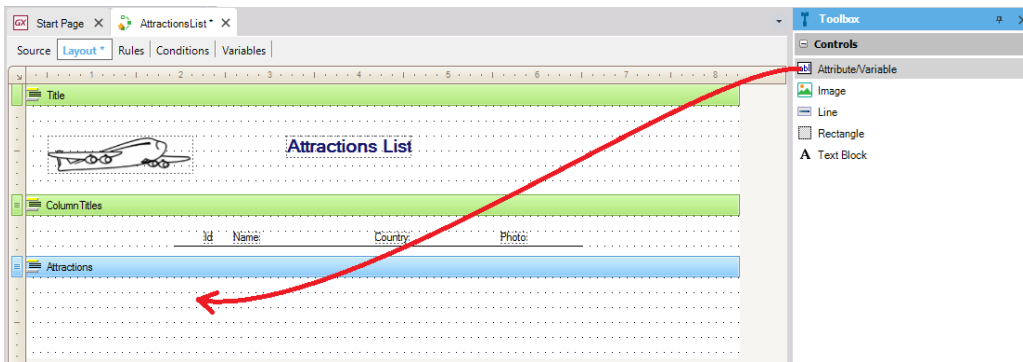


Nos está faltando ahora agregar el tercer printblock que habíamos dicho, para mostrar los datos de las atracciones turísticas, así que insertamos un nuevo printblock... y le ponemos como nombre: Attractions.

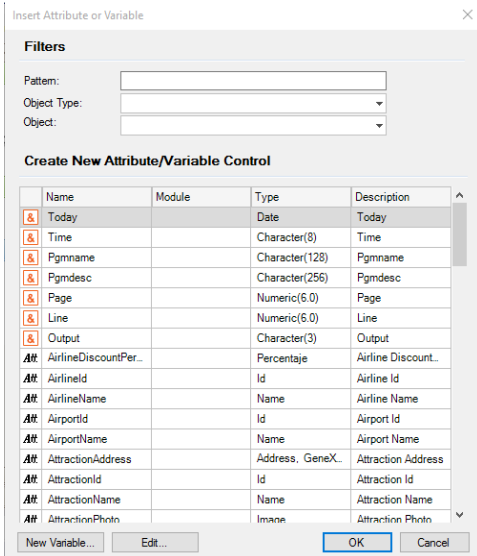


Como los datos están almacenados en atributos, vamos nuevamente a la Toolbox, seleccionamos un control del tipo "Attribute/Variable"

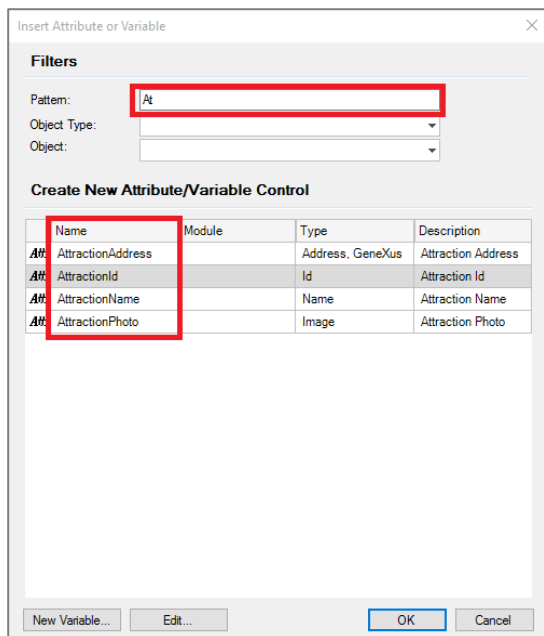
y lo arrastramos bajo el título "Id"...



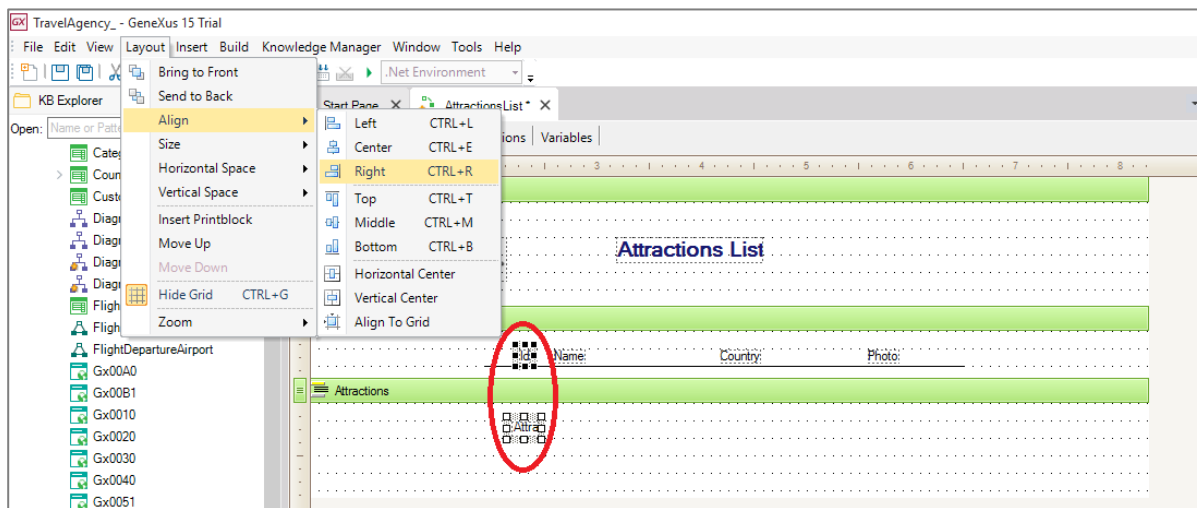
En esta ventana que se nos abre elegimos qué variable o atributo queremos mostrar en el control. Veamos que además de &today, en un procedimiento existen estas otras variables del sistema.



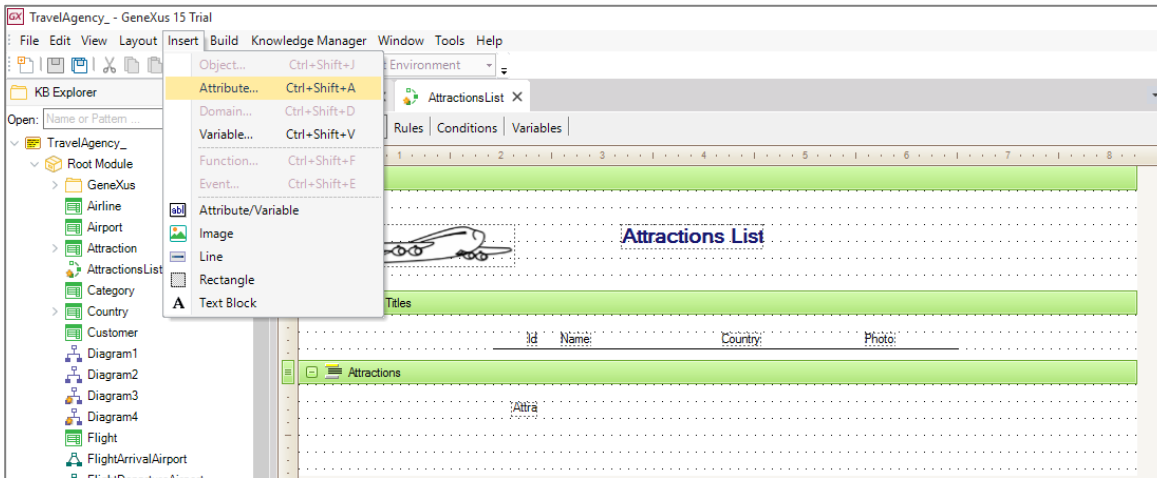
...queremos que muestre el atributo "AttractionId", así que filtramos por "At" y así lo encontramos con facilidad:



Lo alineamos a la derecha respecto al título "Id"... (El último control seleccionado es el que sirve de referencia: vemos que así aparece resaltado gráficamente

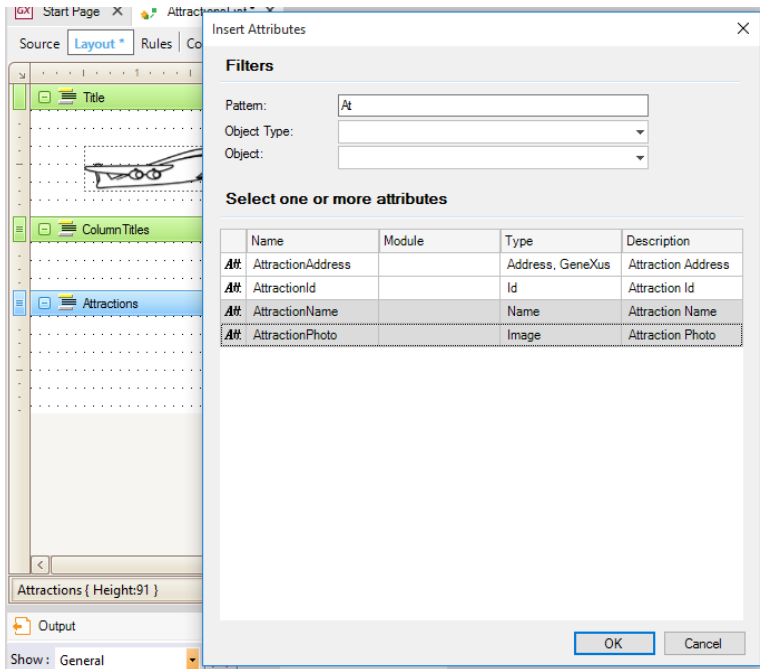


También podemos insertar atributos en un printblock desde la opción **Insert / Attribute**



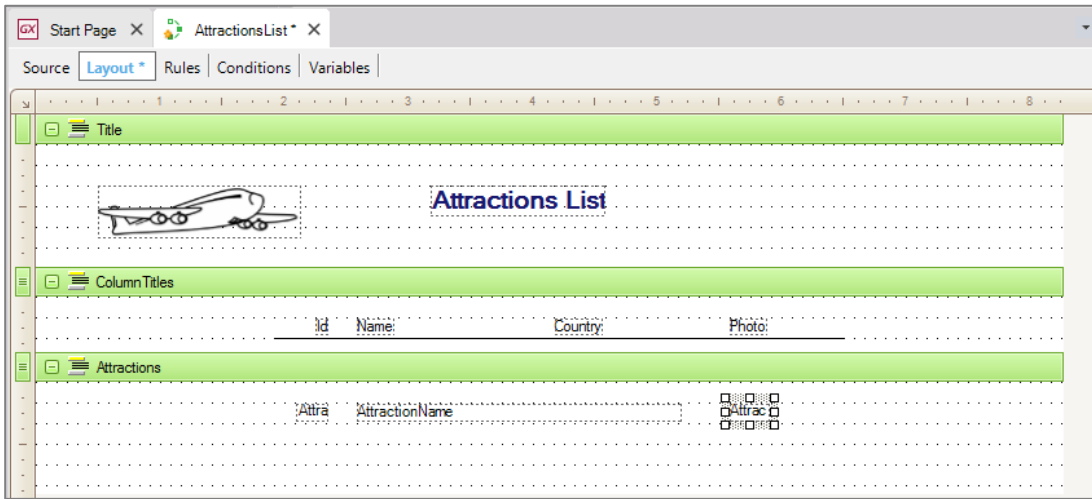
Aquí sólo se ofrecen atributos, no variables. En este cuadro podemos elegir varios atributos a la vez.

Así que seleccionamos AttractionName presionamos Control, y AttractionPhoto.

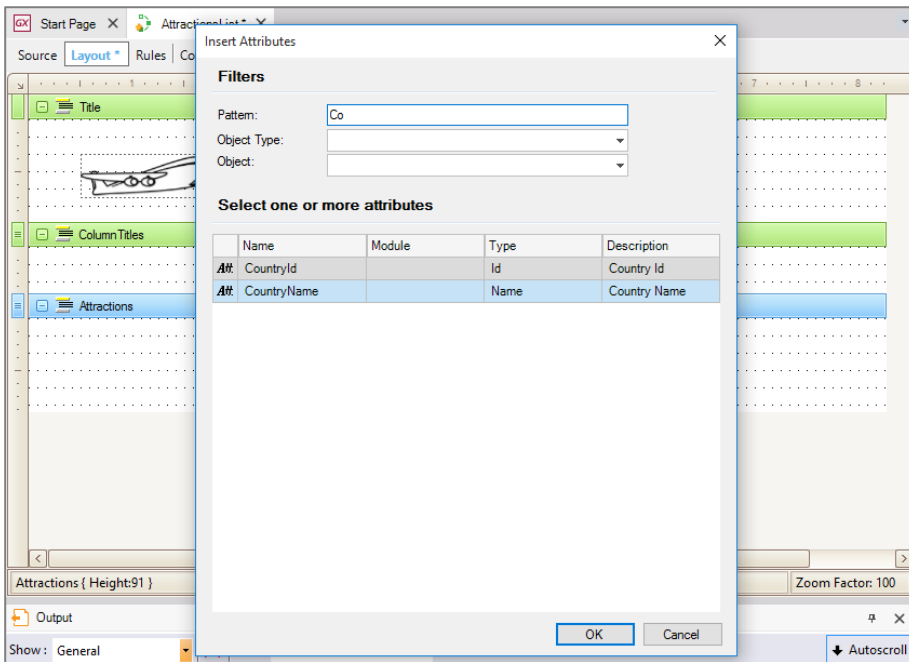


Presionamos OK...

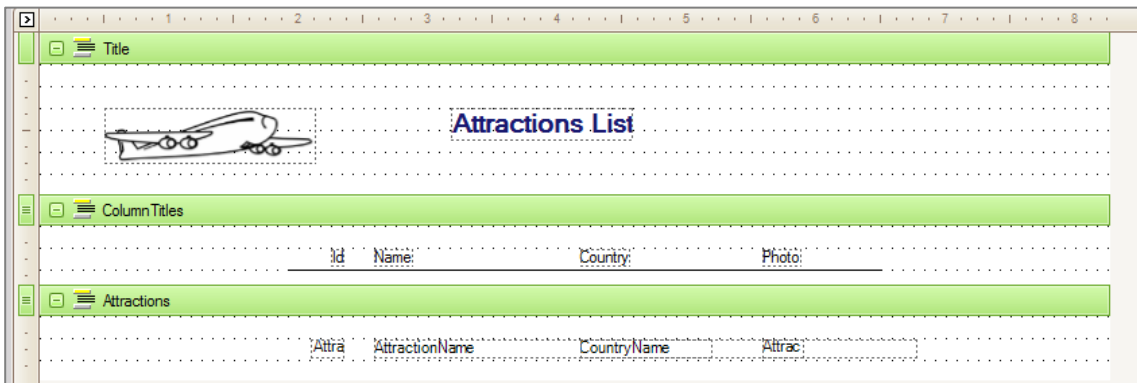
Ubicamos los atributos debajo de los títulos...



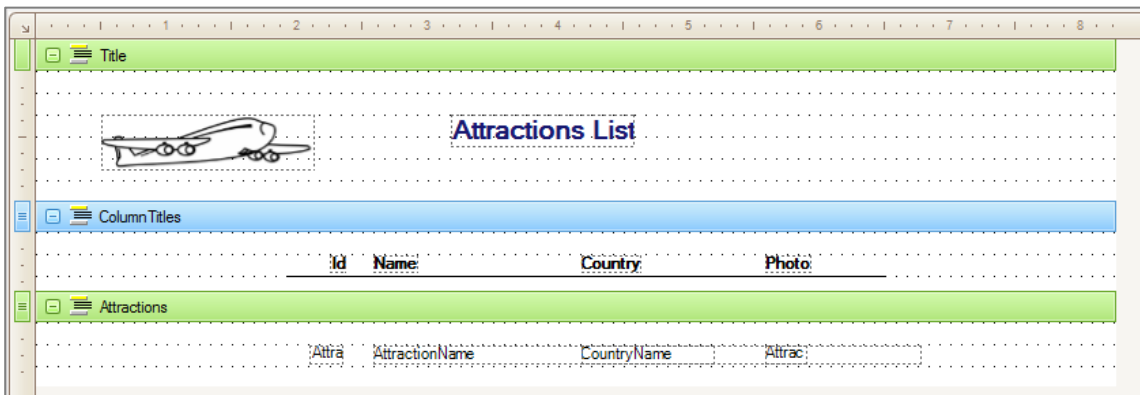
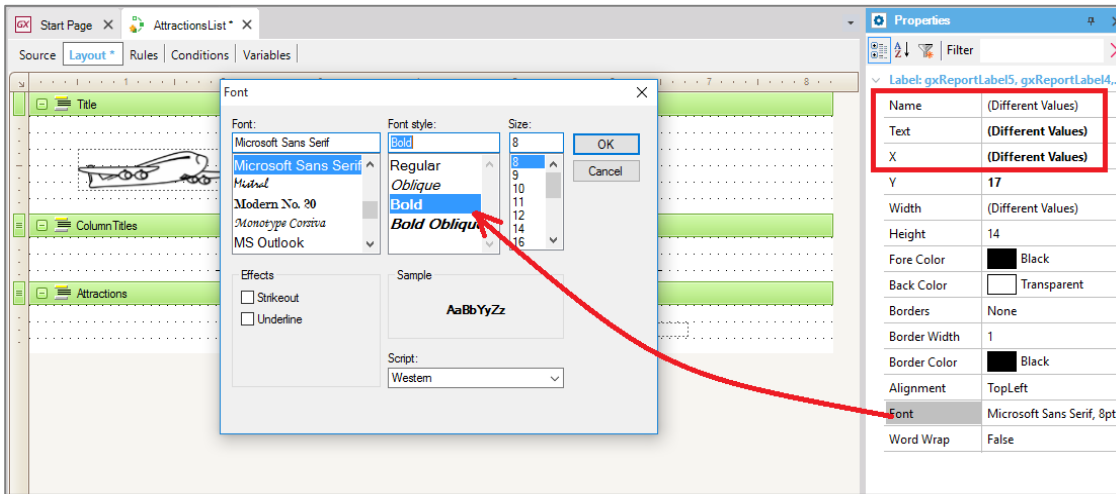
E insertamos CountryName, de igual forma:



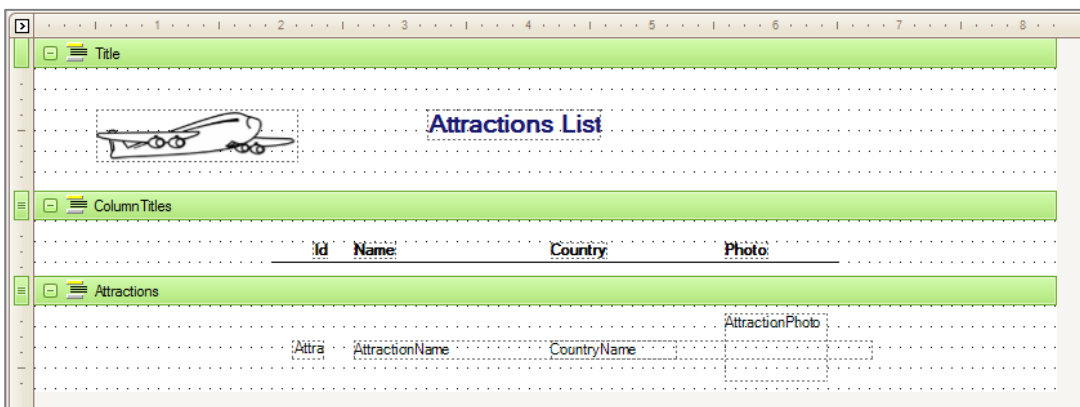
Alineamos los atributos . Y quitamos los espacios en blanco del printblock.



Vamos a poner en negrita los títulos de las columnas, para que se destaquen. Los seleccionamos a todos a la vez, y en las propiedades vemos cómo podemos aplicar de una vez el cambio al Font.



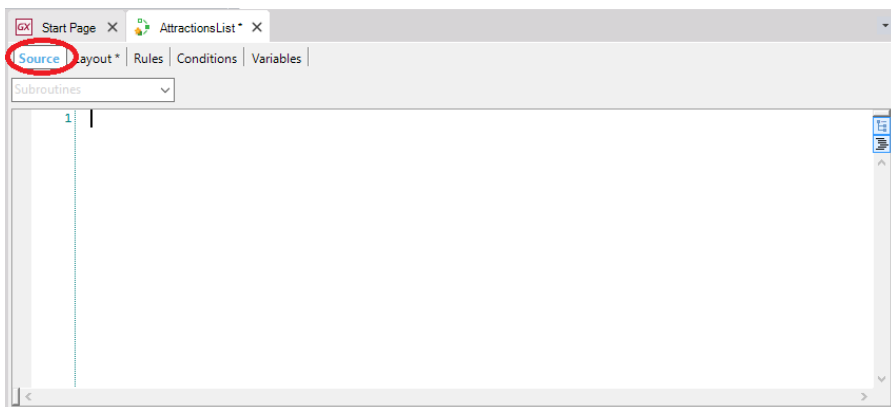
No nos dimos cuenta de una cosa: la foto de la atracción no podrá aparecer en ese espacio tan reducido que le dejamos al control atributo AttractionPhoto. Tendremos que agrandarlo un poco.



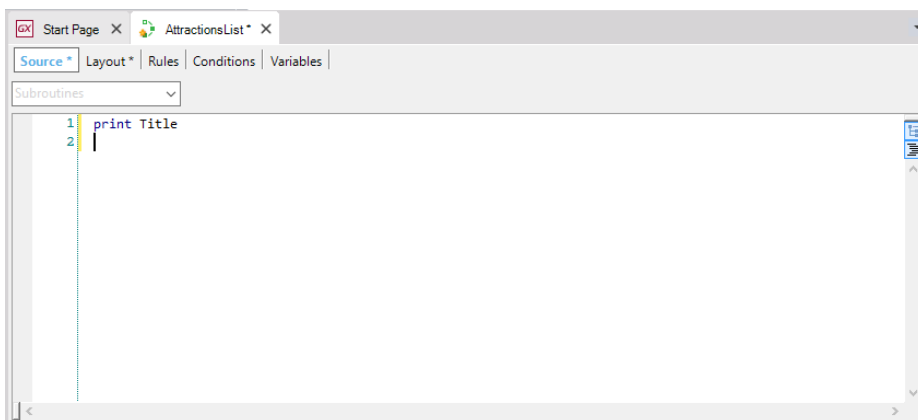
Ya tenemos listo el diseño de cómo queremos que se despliegue la información en el listado.

Ahora falta escribir el código necesario para obtener la información apropiada de la base de datos e ir indicando que se impriman los printblocks en el orden que deseamos.

Vamos entonces a la opción Source...

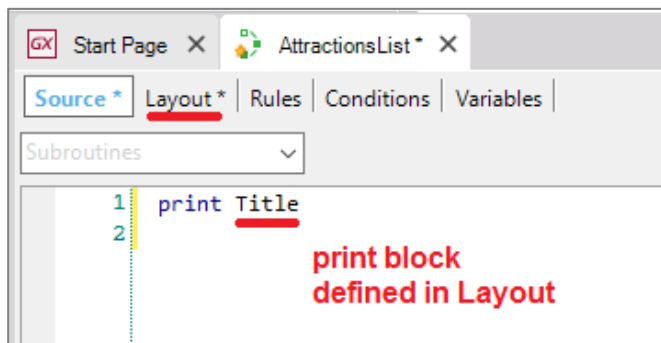


Y lo primero que queremos que se imprima, es el título del reporte, así que escribimos “print Title”:



Como las instrucciones que escribamos en el Source, se ejecutarán de arriba hacia abajo, **ésta instrucción será la primera que se va a ejecutar**. Estamos indicando con ella que se imprima el contenido del printblock de nombre Title, es decir el título del listado.

El comando Print siempre debe tener a continuación el nombre de un printblock definido en el Layout.



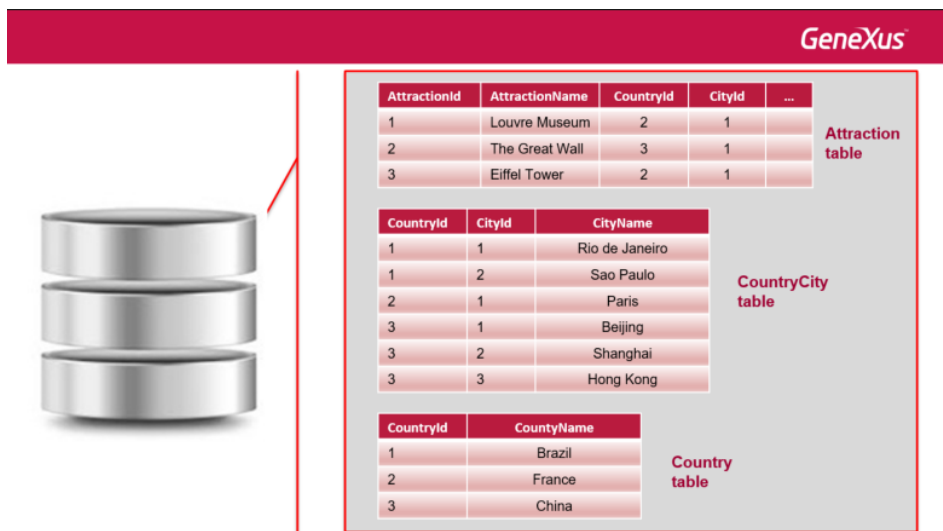
Lo siguiente que deseamos, es imprimir los títulos de las columnas, así que tenemos que dar la orden de imprimir el printblock "ColumnTitles" ...
 Así que escribimos "print... ColumnTitles":

```

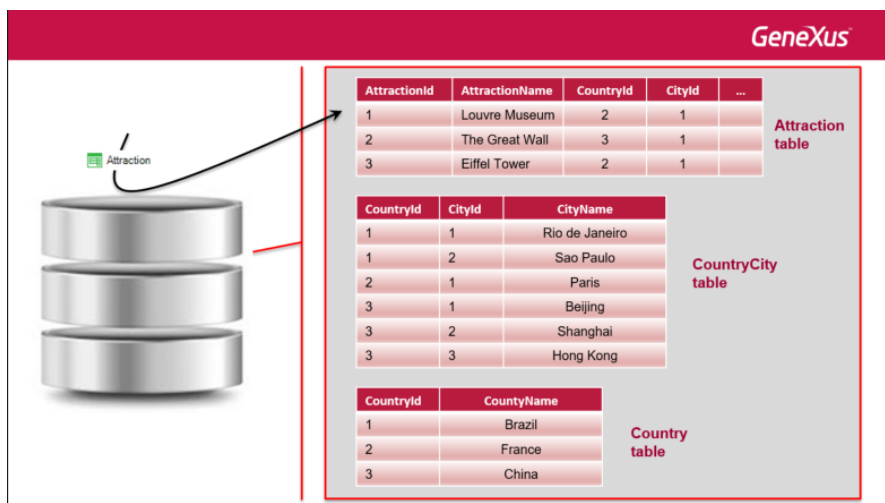
1 print Title
2 print ColumnTitles
    
```

Con estas dos instrucciones hemos indicado que se imprima la parte fija del reporte, es decir, la que no variará de acuerdo a los datos: la que tiene el título del reporte y la imagen del avión y la que tiene los títulos de las columnas.

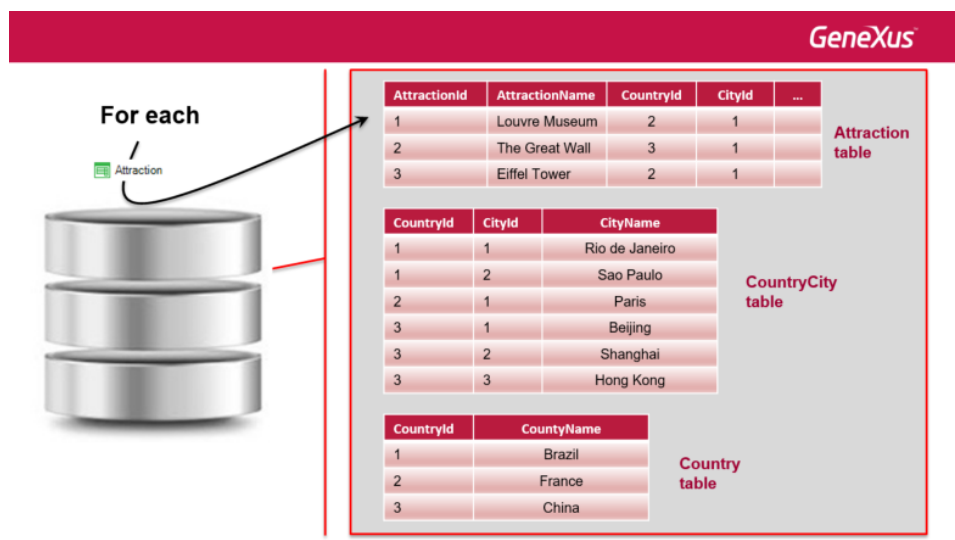
Pero ahora necesitamos imprimir la información de las atracciones, que está almacenada en la base de datos.



Para eso, debemos acceder a la tabla física que tiene almacenada dicha información, o sea a la tabla asociada a la transacción Attraction



El comando que nos permite acceder a una tabla física, es el comando “For Each”. La tabla física accedida se denomina tabla base del for each.



Escribimos entonces el comando For Each...

```

GX Start Page X AttractionsList* X
Source* Layout* Rules Conditions Variables
Subroutines
1 print Title
2 print ColumnTitles
3 For each |

```

y al lado: Attraction.

```

GX Start Page X AttractionsList* X
Source* Layout* Rules Conditions Variables
Subroutines
1 print Title
2 print ColumnTitles
3 For each Attraction|

```

¿Por qué escribimos Attraction al lado de For each?
 porque es el nombre de la transacción **cuya tabla física asociada** queremos navegar...

.. y ahora.. dado que queremos imprimir de cada atracción turística, el contenido de los atributos AttractionId, AttractionName, CountryName, y AttractionPhoto escribimos la orden para imprimir el printblock “Attractions” que los contiene. Así que escribimos Print Attractions:

```

1 print Title
2 print ColumnTitles
3 For each Attraction
4   print Attractions

```

y luego cerramos al comando For Each con la instrucción Endfor

```

1 print Title
2 print ColumnTitles
3 For each Attraction
4   print Attractions
5 endfor

```

¡Y listo!

De esta forma, le hemos indicado a GeneXus que tiene que navegar la tabla física ATTRACTION, correspondiente a la transacción Attraction:

For each

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

Attraction table

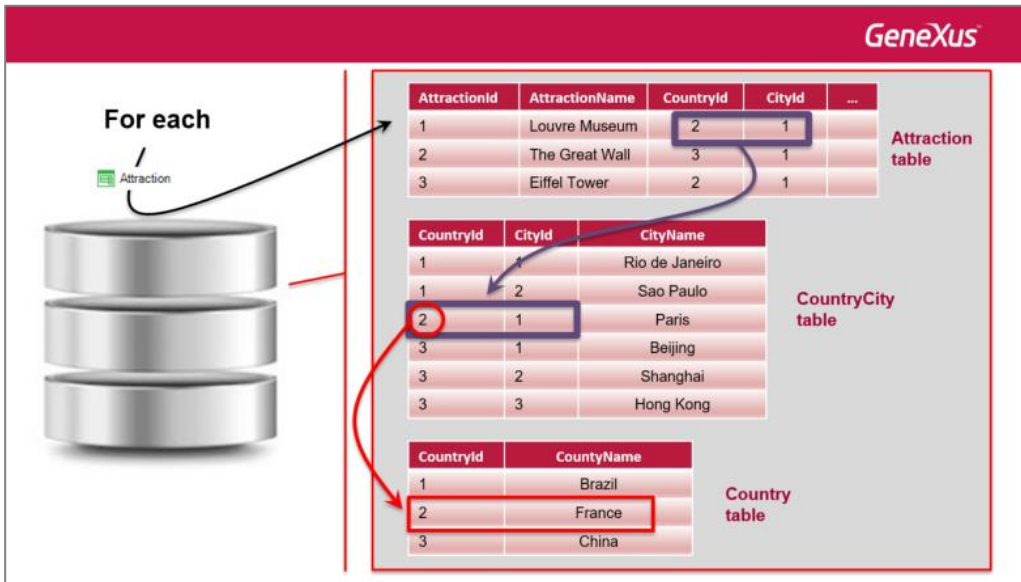
CountryId	CityId	CityName
1	1	Rio de Janeiro
1	2	Sao Paulo
2	1	Paris
3	1	Beijing
3	2	Shanghai
3	3	Hong Kong

CountryCity table

CountryId	CountryName
1	Brazil
2	France
3	China

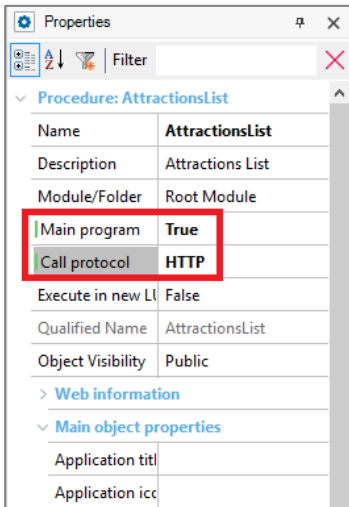
Country table

y como dentro del For each, hemos invocado a un printblock que contiene atributos de las tablas ATTRACTION y COUNTRY, aplicando el concepto de tabla extendida, para cada atracción navegada, se accederá a la tabla COUNTRYCITY, y de ésta a COUNTRY, para obtener el nombre del país donde se encuentra dicha atracción:

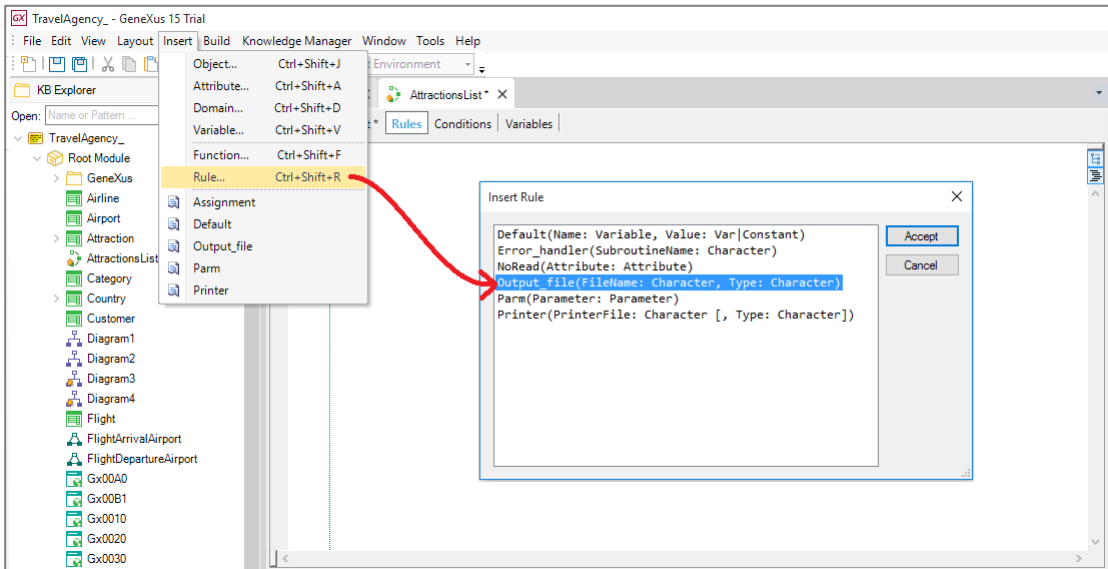


Vamos entonces a ejecutar para ver el resultado.

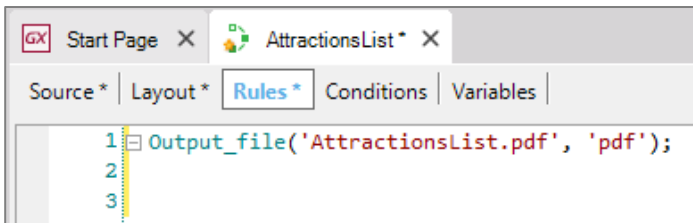
Pero primero tenemos que definir algunas propiedades necesarias para que se imprima el listado con formato PDF. [para las props usar vista categorizada, no alfabética que apabulla!!!]Vamos a las propiedades del reporte y en la propiedad “Main program” seleccionamos True. Luego en la propiedad “Call protocol” seleccionamos “HTTP”.



Y por último tenemos que insertar la regla OutputFile en la sección de las reglas...porque como se ve, un objeto de este tipo también permite definir algunas reglas –aunque menos que en una transacción–, así que seleccionamos Insert/Rule

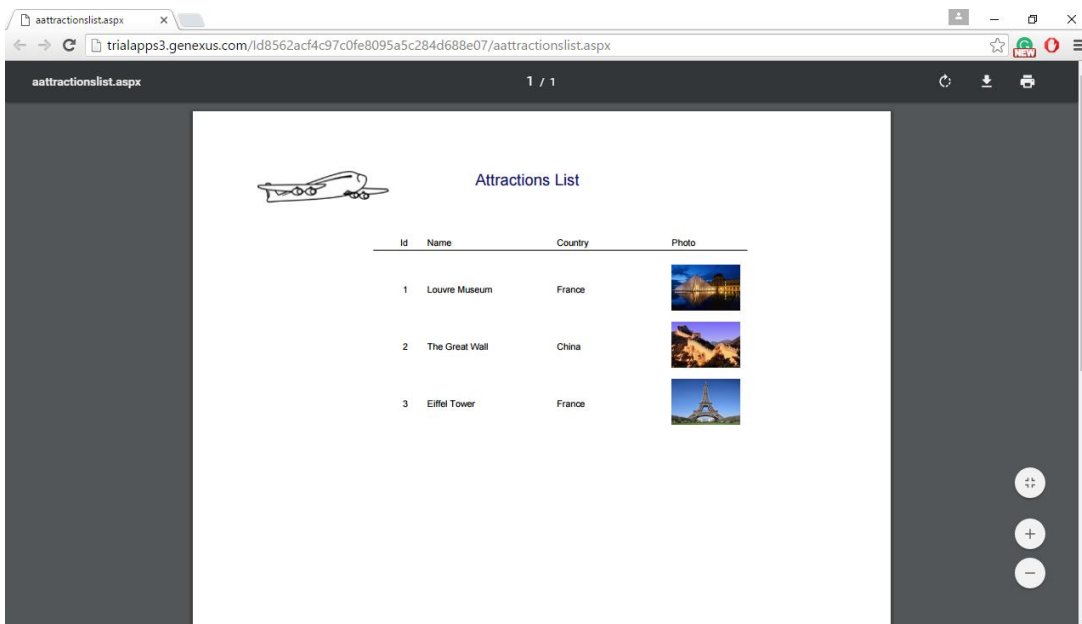


Y completamos poniendo el nombre del archivo del listado “AttractionsList.PDF” y luego el formato que vamos a utilizar: “PDF”.



Salvamos... y ahora ya podemos ejecutarlo.

¡y vemos que se crea el listado! ... con el formato que definimos... y aparecen listadas todas las atracciones turísticas que habíamos ingresado, cada una de ellas con el nombre del país al que pertenece y la foto.



Volvamos a GeneXus

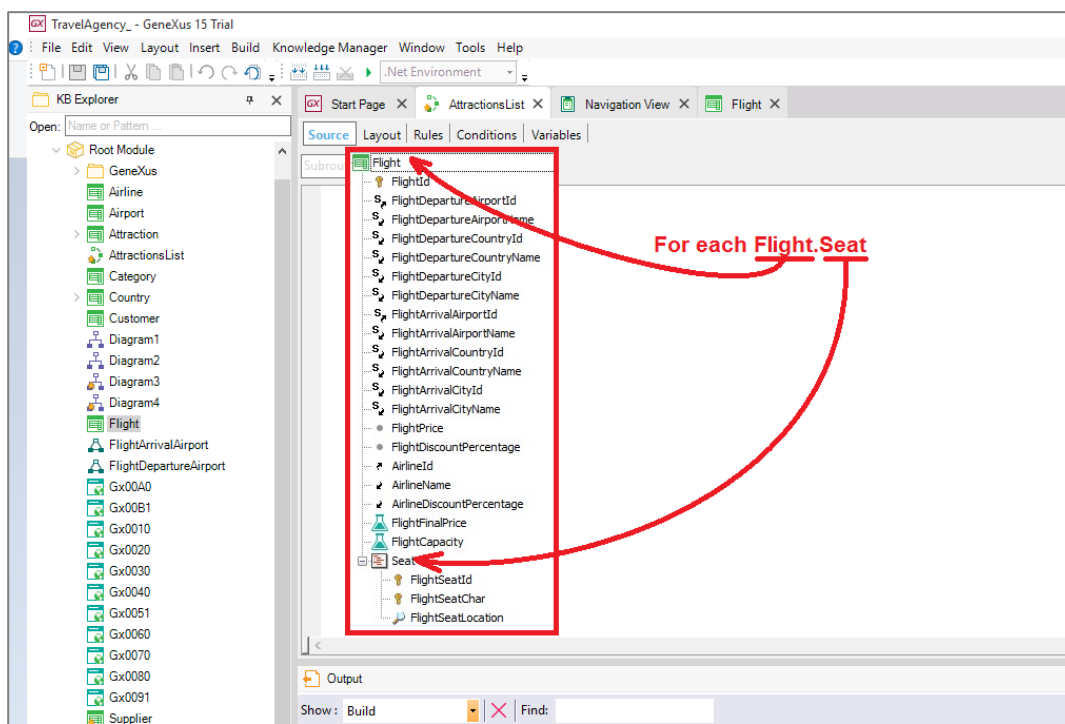
Aquí denominamos a Attraction como: **Base transaction**

For each Attraction  **BASE TRANSACTION**
 Print Attractions
Endfor

y como dijimos antes, corresponde al nombre de la transacción, cuya tabla física asociada queremos navegar.
En nuestro caso, la tabla Attraction, de igual nombre que la transacción.

¿Y si la transacción tuviera más de un nivel? Como Flight por ejemplo... y quisiéramos navegar la tabla física asociada al 2do nivel de la transacción? Es decir: a los asientos del vuelo.

La sintaxis en ese caso sería la siguiente:

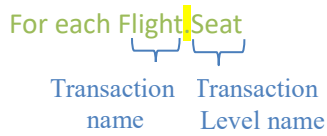


O sea: el nombre de la transacción:

For each Flight.Seat
 └─┬─┘
 Transaction
 name

Punto

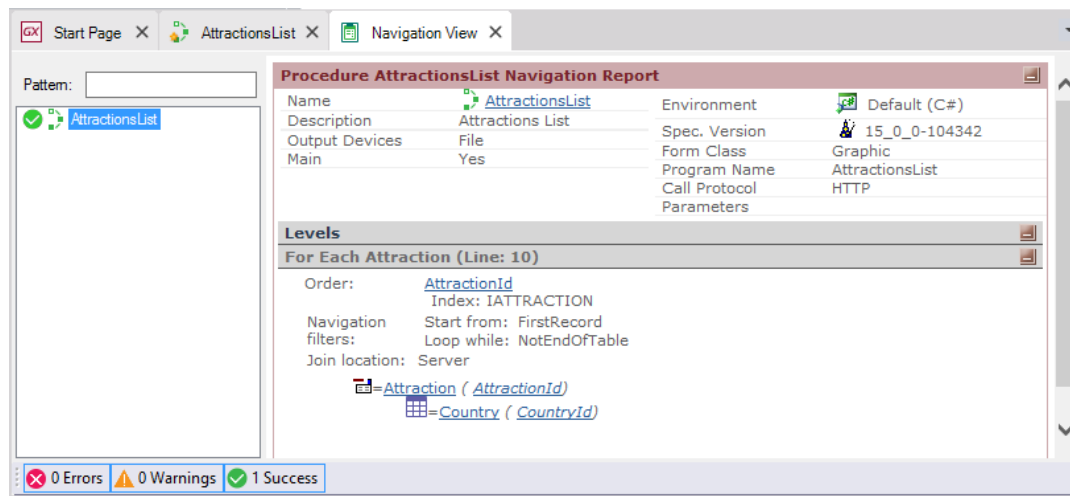
Y pegado al punto, el nombre del nivel:



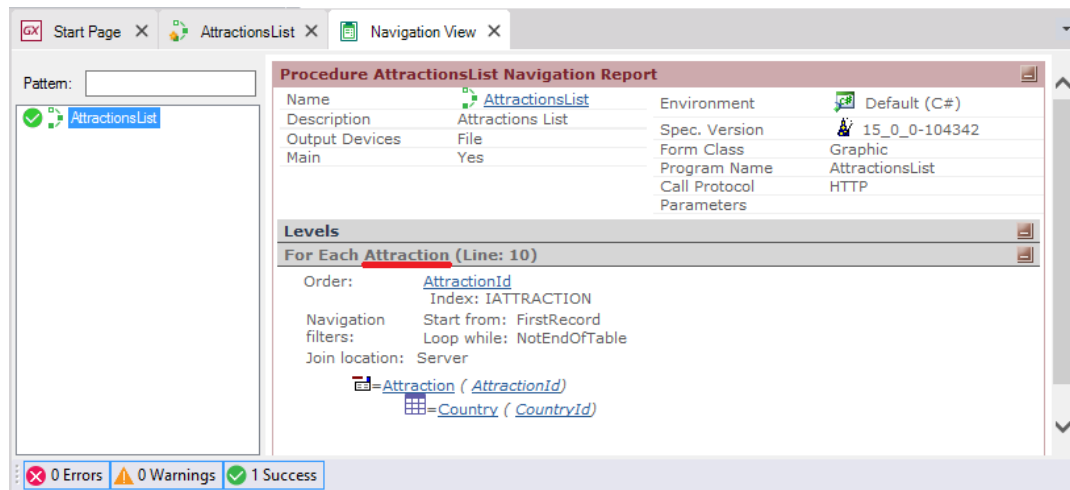
Bien. La **tabla física que el For each navegará**, así como otras decisiones que toma GeneXus, son mostradas en el **listado de navegación del procedimiento**

Este listado es creado automáticamente cuando se genera el procedimiento para ser ejecutado. En nuestro caso, fue luego del F5.

En él GeneXus nos indica cómo accede a la información de la base de datos. Vamos a observarlo

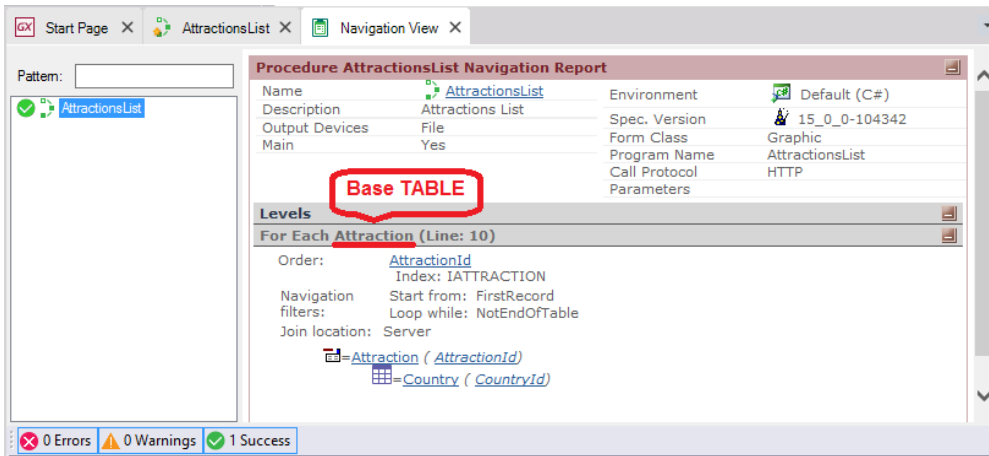


Podemos ver que al lado de donde dice "For Each", dice también Attraction:

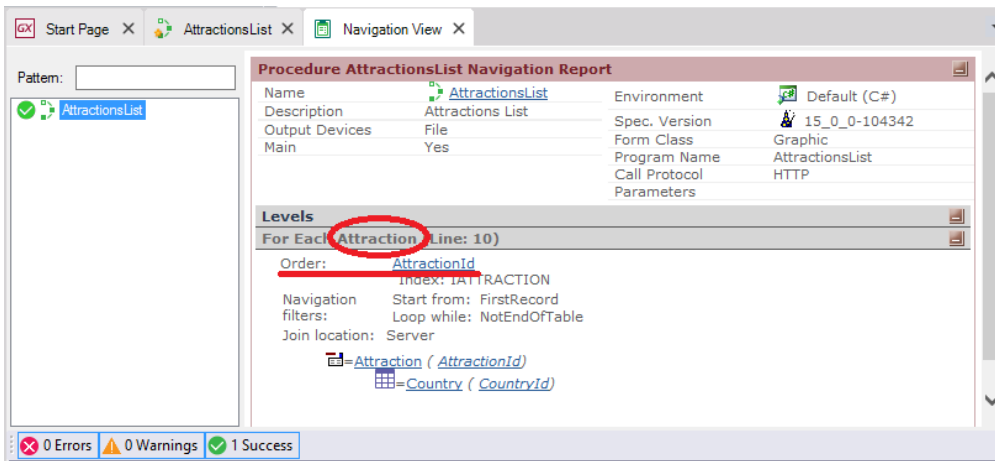


para indicarnos que ésta es la **tabla base del For Each**.

Recordemos que el For each recorre una tabla física, por lo que el nombre Attraction que aparece en el listado de navegación es el de la tabla física ATTRACTION, no el de la transacción base que hemos escrito en el procedimiento. GeneXus deduce esta tabla porque es la tabla asociada a esa transacción base que indicamos.

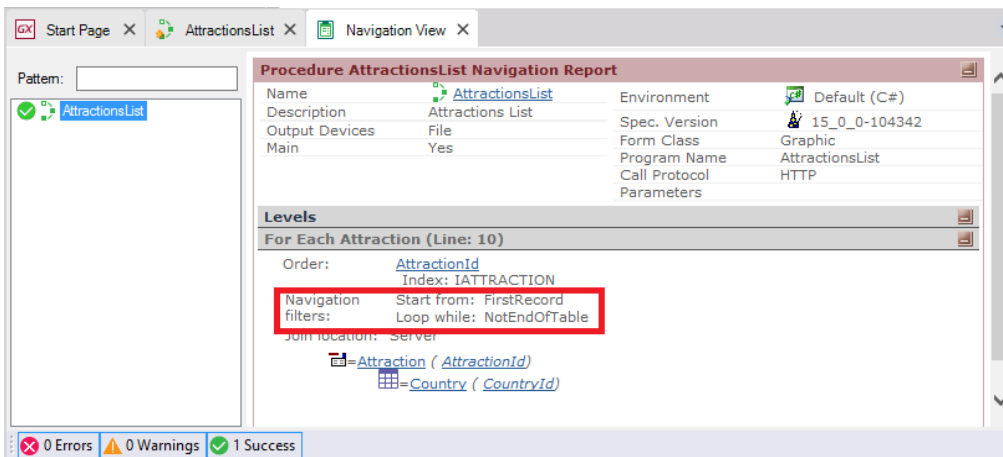


También nos indica que para ordenar el listado de atracciones

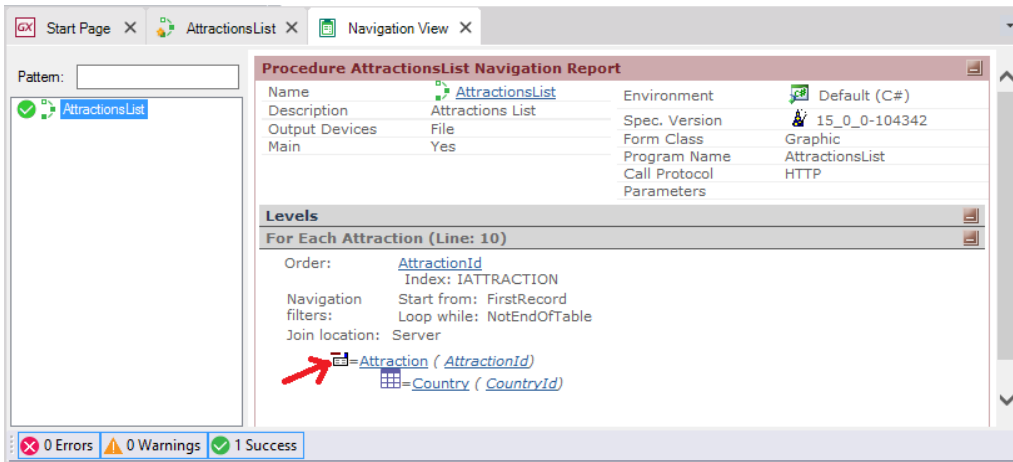


se utilizó el atributo AttractionId (que es la clave primaria de la tabla Attraction)...

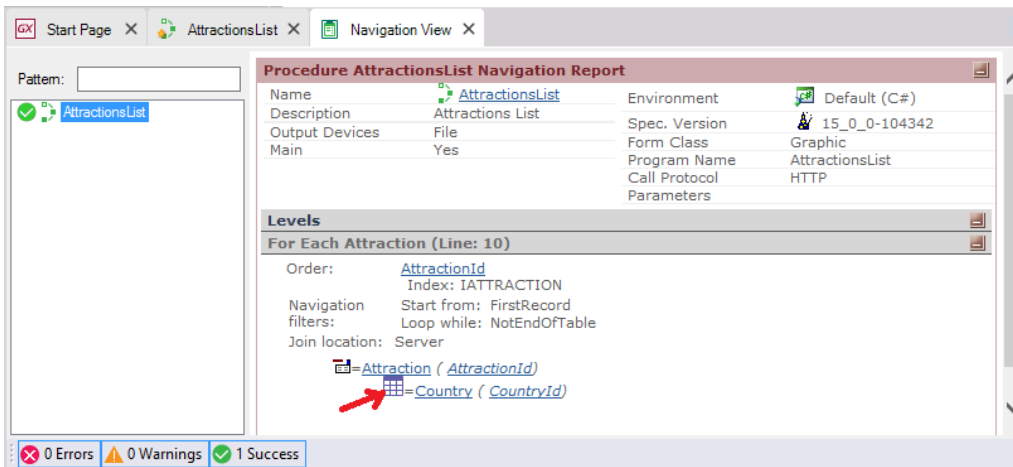
Nos indica también que recorrió todos los registros de la tabla: pues empezó por el primer registro e iteró hasta llegar al fin de la tabla.



es decir que mostró todas las atracciones.... y por último... nos indica que la tabla que navegó fue Attraction



y que **debió acceder** a Country



para recuperar información, ya que en nuestro listado mostramos al nombre del país.

Attractions List

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

Volvamos ahora al Source del procedimiento

Una cosa que nos había quedado pendiente es que las atracciones salgan listadas ordenadas en forma alfabética,




o sea por nombre de atracción.

Y esto lo logramos simplemente escribiendo luego del “For Each Attraction”, la cláusula “order AttractionName”

```
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
```

Vamos ahora a ejecutar el procedimiento para verlo...

Y vemos que ahora las atracciones turísticas son listadas ordenadas alfabéticamente por su nombre.

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

Si observamos el listado de navegación, vemos cómo lo indicaba:

Procedure AttractionsList Navigation Report

Name	AttractionsList	Environment	Default (C#)
Description	Attractions List	Spec. Version	15_0_0-104342
Output Devices	File	Form Class	Graphic
Main	Yes	Program Name	AttractionsList
		Call Protocol	HTTP
		Parameters	

Warnings

- spc0038 There is no index for order [AttractionName](#); poor performance may be noticed in group starting at line 3.

Levels

For Each Attraction (Line: 10)

Order: [AttractionName](#)
! No index

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

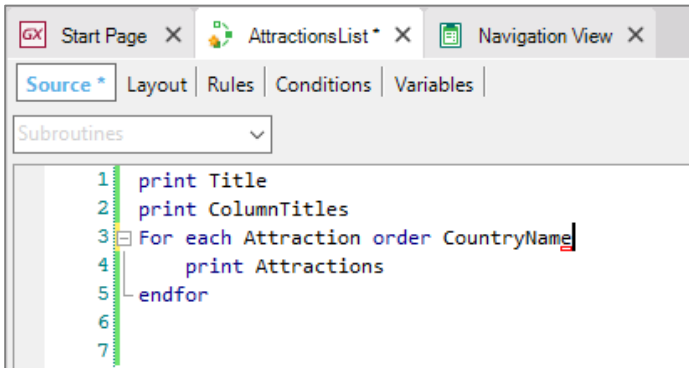
[Attraction](#) ([AttractionId](#)) INTO [CountryId](#) [AttractionPhoto.Uri](#)
[AttractionPhoto](#) [AttractionId](#) [AttractionName](#)
[Country](#) ([CountryId](#)) INTO [CountryName](#)

0 Errors 0 Warnings 1 Success

Por ahora no prestemos atención a la advertencia que el listado nos informa.

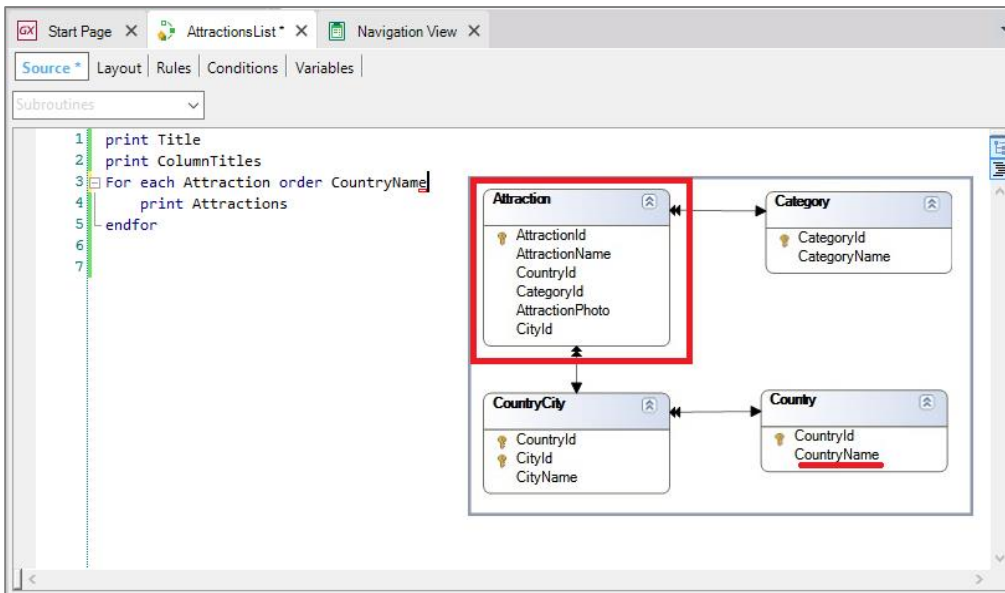
Ahora vamos a observar algo interesante. Y es que GeneXus nos permite ordenar por el valor de un atributo que no está en la tabla ATTRACTION, sino en su tabla extendida.

Cambiamos el atributo del orden, por CountryName:



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

Este atributo no se encuentra físicamente en la tabla base del For each



pero se encuentra en la tabla extendida de la tabla base navegaday por lo tanto podemos ordenar por él.

Ejecutemos el procedimiento para ver el resultado.

Y vemos que ¡ahora salen listadas las atracciones ordenadas alfabéticamente por el nombre de país!

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

Así como hemos agregado la **cláusula opcional "order"**, al For each, la sintaxis del For each permite que le agreguemos varias cláusulas y definiciones opcionales más, como veremos.

Por ejemplo, ¿qué pasaría si en la agencia de viajes nos piden que listemos solamente las atracciones turísticas de Francia?

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

Solamente agregaremos al comando For Each, una cláusula llamada **Where**, para que filtre y muestre únicamente los datos que cumplan con la condición deseada.


Así que nos posicionamos en el renglón siguiente al For Each y escribimos Where...CountryId=2, ya que sabemos que el Id de Francia es el 2.

```



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

```

Hagamos F5.



Attractions List


Id	Name	Country	Photo
1	Louvre Museum	France	
3	Eiffel Tower	France	

En lugar de filtrar por el identificador de país, podríamos haber 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 y ejecutamos otra vez el reporte....

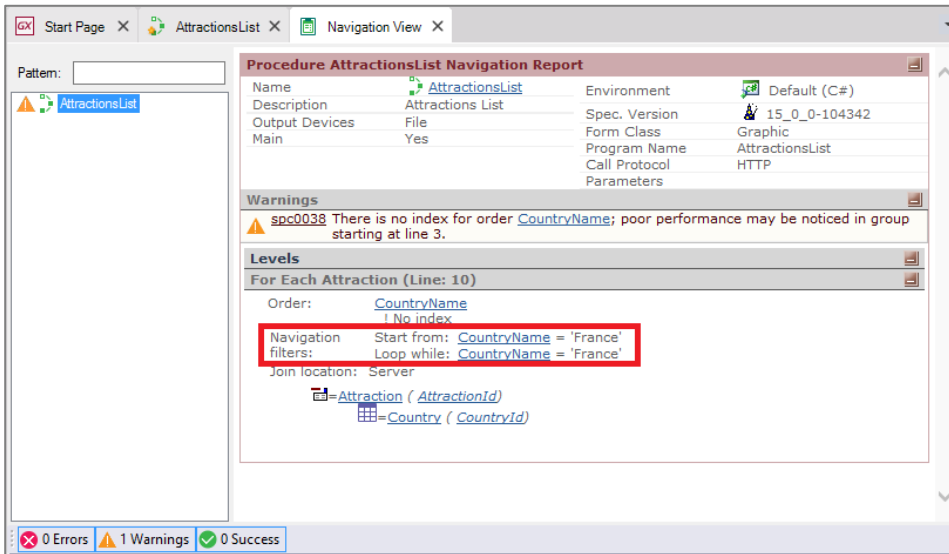
¡Y vemos que el resultado es exactamente el mismo!



Attractions List

Id	Name	Country	Photo
1	Louvre Museum	France	
3	Eiffel Tower	France	

Si observamos ahora el listado de navegación:

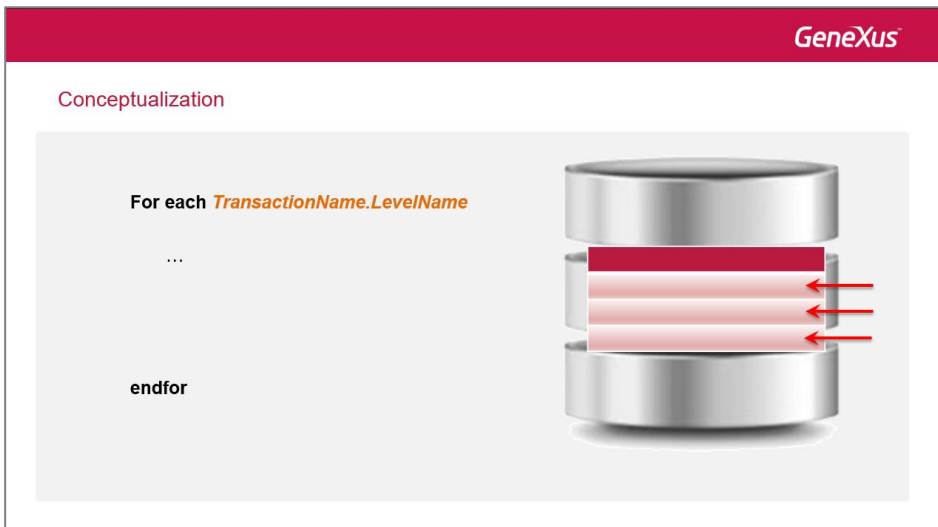


Vemos que ya no se recorre toda la tabla Attractions. Como estamos ordenando por CountryName, para quedarnos con los países de nombre 'France' solamente tiene que recorrer un pedacito de la tabla y no toda. Es análogo a buscar en un diccionario la palabra 'France'. No se busca en todo el diccionario. Se accede directamente a la "F".

Finalmente actualizamos los cambios en GeneXus Server.

Repasemos ahora los conceptos aprendidos:

El comando For each se utiliza para recorrer cada registro de una tabla y hacer algo con su información relacionada.



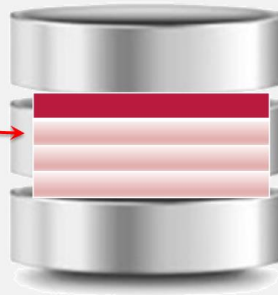
Para ello le indicamos el nombre del nivel de la transacción cuya tabla asociada queremos recorrer.

Conceptualization

For each TransactionName.LevelName

...

endfor



A esta indicación de nivel le llamamos **transacción base** del for each:

Conceptualization

For each **Base transaction**
TransactionName.LevelName

...

endfor



Y de ese nivel GeneXus inferirá la tabla a recorrer, a la que llamamos **tabla base** del for each.

Conceptualization

For each **Base transaction**
TransactionName.LevelName

...

endfor



El conjunto de atributos que están entre el For each y el Endfor deben pertenecer a la tabla extendida de esa tabla base a recorrer.

GeneXus

Conceptualization

```
For each TransactionName.LevelName  
  ...  
endfor
```

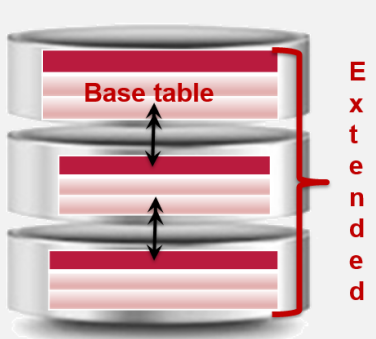


Diagram illustrating the relationship between a base table and an extended table. The base table is shown as the top cylinder, and the extended table is shown as the middle and bottom cylinders. The extended table is labeled 'Extended' on the right. The base table is labeled 'Base table'.

Aquí resumimos lo que hemos visto hasta el momento del comando for each:

GeneXus

Conceptualization

```
For each BaseTransaction  
  order Att1, Att2, ..., Attn  
  where condition  
    MainCode  
endfor
```

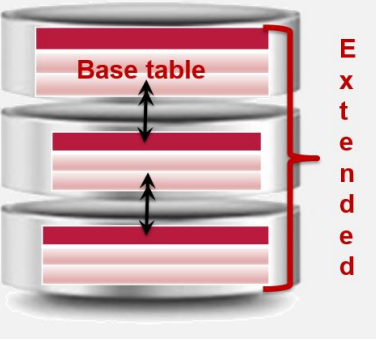
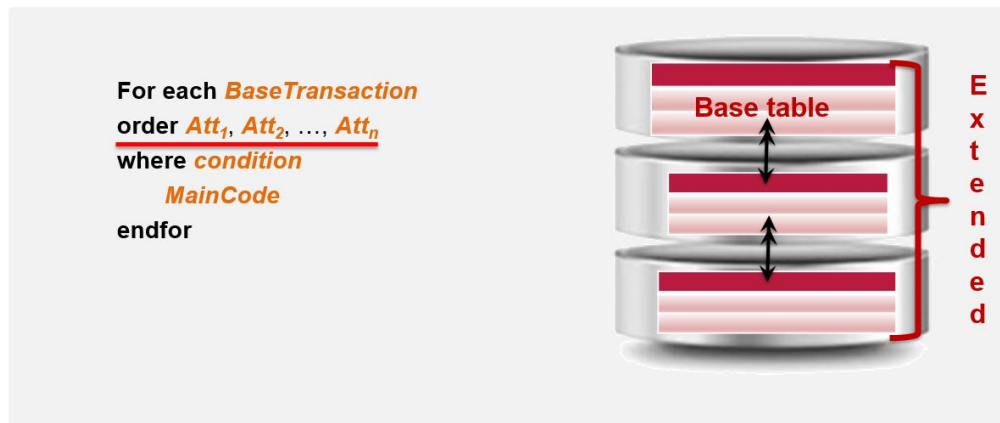


Diagram illustrating the relationship between a base table and an extended table. The base table is shown as the top cylinder, and the extended table is shown as the middle and bottom cylinders. The extended table is labeled 'Extended' on the right. The base table is labeled 'Base table'.

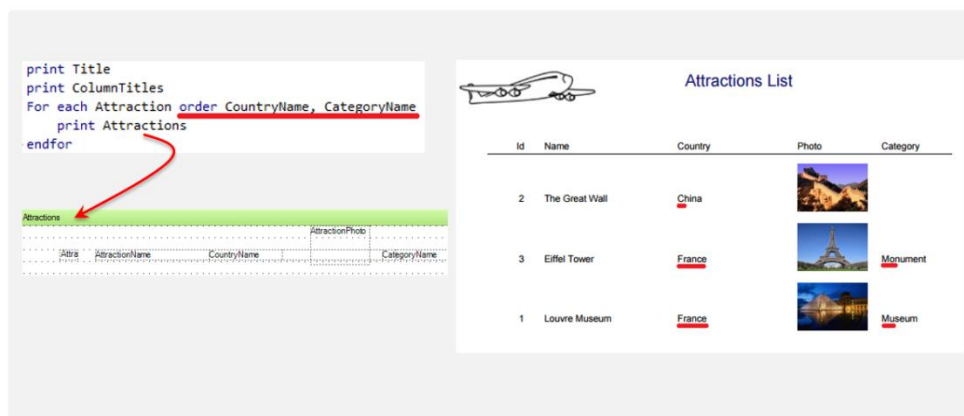
Mediante la cláusula **Order** es posible indicar el criterio por el cual ordenar la información devuelta por el For each. El orden puede ser de acuerdo a los atributos de la tabla base del For each o de su extendida. Como vemos, se puede ordenar por un solo atributo, o por varios.

Conceptualization



Por ejemplo, si también mostráramos la categoría de la atracción turística en el listado y quisiéramos ordenarlo por nombre de país y dentro de las atracciones que son del mismo país, por nombre de categoría... escribiríamos ambos atributos en forma ordenada: primero CountryName y luego CategoryName...

Conceptualization



Aquí tanto CountryName, como CategoryName no están presentes en la tabla base, Attraction, sino en tablas de la extendida:

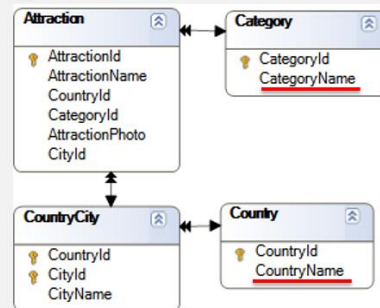
Conceptualization

```

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

```

Attractions				
AttractionId	AttractionName	CountryName	AttractionPhoto	CategoryName
...



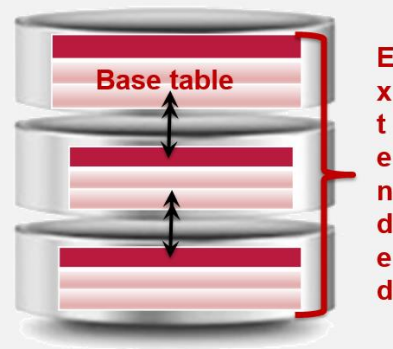
Para filtrar la información devuelta por el For each se utiliza la cláusula **Where**, en la que especificamos la condición que deberán cumplir los registros para ser elegidos:

Conceptualization

```

For each BaseTransaction
order Att1, Att2, ..., Attn
where condition
  MainCode
endfor

```



La condición puede ser compleja, incluyendo varias condiciones unidas por AND u OR, es decir, por ejemplo:

- Condition1 **AND** Condition2: lo que significa que ambas deben cumplirse a la vez

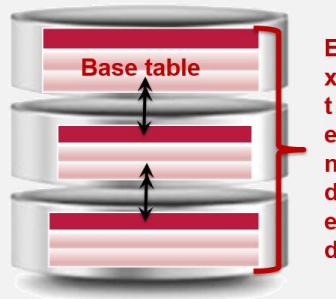
Conceptualization

```

For each BaseTransaction
order Att1, Att2, ..., Attn
where condition
      MainCode
endfor

```

condition₁ and *condition₂*



- Condition1 **OR** Condition2: lo que significa que de cumplirse una de las dos, ya alcanza para que el registro que se está evaluando pase el filtro:

Conceptualization

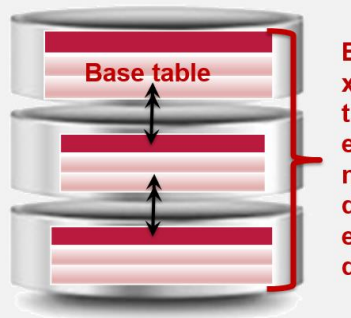
```

For each BaseTransaction
order Att1, Att2, ..., Attn
where condition
      MainCode
endfor

```

condition₁ and *condition₂*

condition₁ or *condition₂*



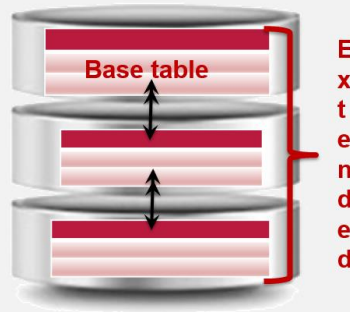
También podemos colocar varias cláusulas **Where**, lo que es lo mismo que escribir una sola, con las condiciones unidas por **AND**:

Conceptualization

```

For each BaseTransaction
order Att1, Att2, ..., Attn
where condition1 } and
where condition2 } and
... }
where conditionn } and
MainCode
endfor

```



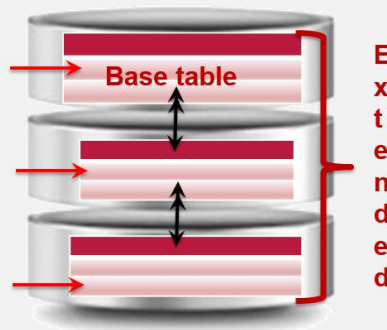
Dentro del comando for each, en su **código principal**, se escriben los comandos que queremos que se ejecuten uno a continuación del otro para realizar paso a paso lo que se necesite con el registro de la tabla base sobre el que se está posicionado en cada momento... y los asociados por tabla extendida:

Conceptualization

```

For each BaseTransaction
order Att1, Att2, ..., Attn
where condition1
where condition2
...
where conditionn
MainCode
endfor

```



Por ejemplo, imprimir un printblock:


Conceptualization



```

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

```

Attractions
AttractionName
CountryName
AttractionPhoto
CategoryName

 Attractions List

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

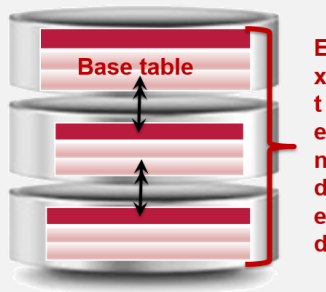
Así nos queda, por tanto, la estructura del comando for each hasta donde vimos:

Conceptualization

```

For each BaseTransaction
  order Att1, Att2, ..., Attn
  where condition1
  where condition2
  ...
  where conditionn
  MainCode
endfor

```



El comando admite más cláusulas y opciones. Alguna será vista en otros videos. Otras se abordarán en otros cursos.

GeneXus™

Videos training.genexus.com
 Documentation wiki.genexus.com
 Certifications training.genexus.com/certifications