

# Procedimientos y Listados.

Comando para consultar la base de datos

*GeneXus™ 16*

## Objeto Procedure

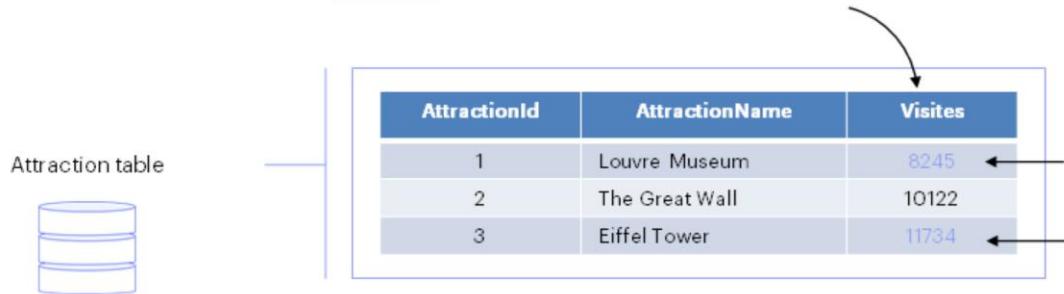
- Permite definir procesos para acceder y navegar tablas de la base de datos con distintos objetivos...



A continuación veremos algunos objetivos posibles.

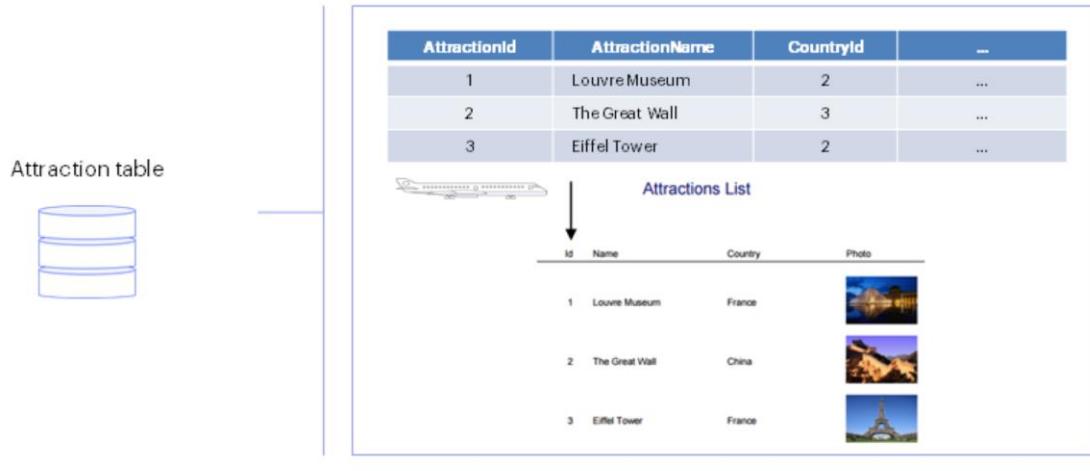
## Objeto Procedure

- Navegar los **registros** de cierta tabla, que cumplan con determinadas condiciones y actualizar para **esos** registros cierto atributo con determinado valor.



## Objeto Procedure

- Navegar cierta tabla e **imprimir** todos sus datos en un listado PDF, ordenados por algún criterio.



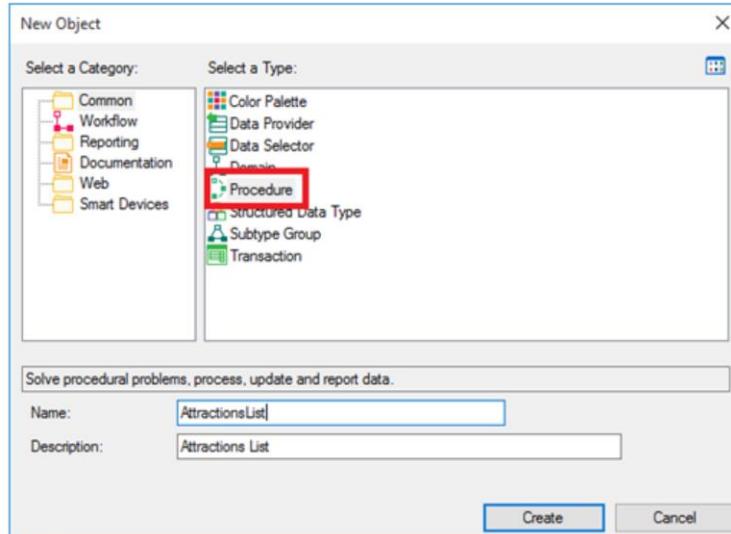
## Objeto Procedure

- Definir procesos que contengan búsquedas, cálculos, actualizaciones a la base de datos y que vayan imprimiendo información.



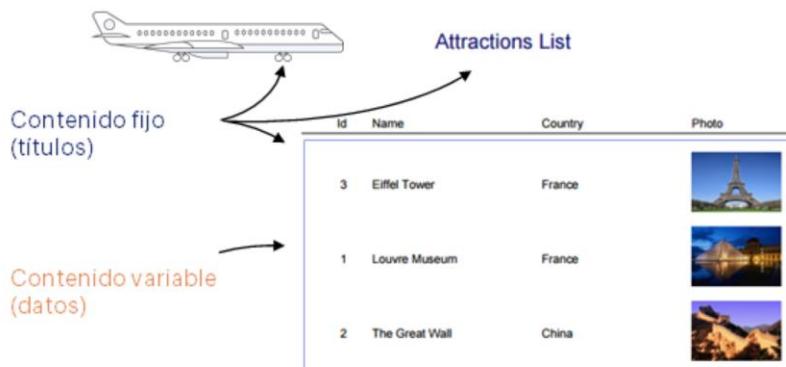
Todas estas cosas las resolvemos creando Procedimientos en nuestra base de conocimiento (objeto GeneXus Procedure).

## Objeto Procedure



## Ejemplo: listar datos en formato PDF

- ✓ Se desea: listar en un archivo PDF todas las atracciones turísticas de la agencia de viajes, ordenadas alfabéticamente.



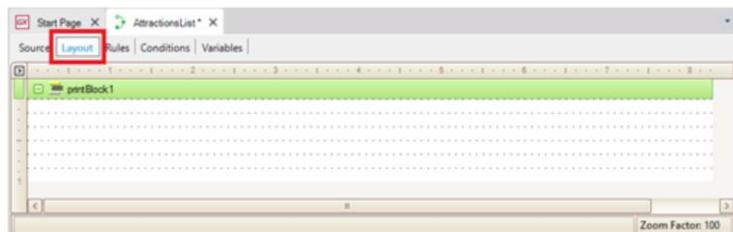
Comencemos definiendo un procedimiento que muestre todas las atracciones turísticas que ofrece la agencia de viajes, en orden alfabético.

## Procedure: Source y Layout

- ✓ Source: para las instrucciones a ser ejecutadas



- ✓ Layout: para diseñar la salida

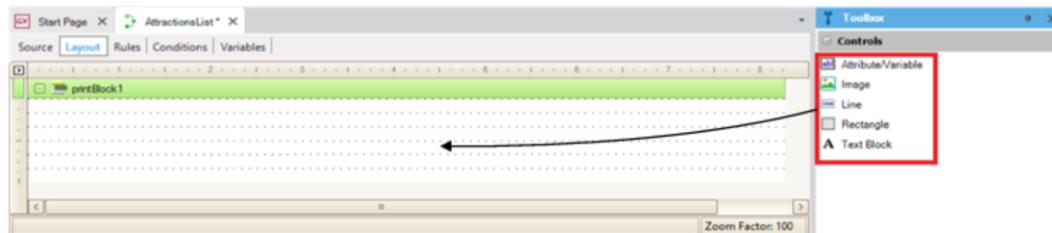


Una vez creado el objeto vemos que GeneXus nos posiciona en una sección llamada Source. Aquí es donde escribiremos comandos y órdenes que permitan al procedimiento cumplir con el objetivo para el cual lo hemos creado, en nuestro caso imprimir un listado de atracciones turísticas.

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.

## Layout

- ✓ Se compone de printblocks



- ✓ Por defecto ya aparece uno, de nombre printBlock1.

El Layout se compone de printblocks y dentro de los printblocks incluiremos lo que deseamos mostrar.

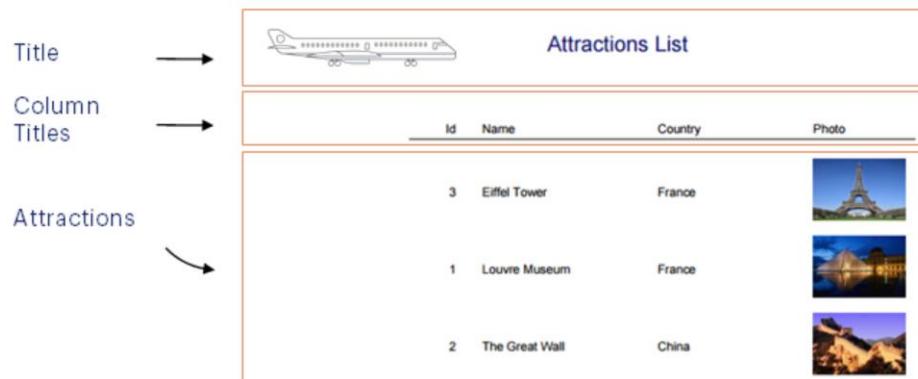
Podemos querer mostrar títulos, líneas, rectángulos, imágenes, así como también valores de atributos o variables. Para ello los arrastraremos dentro del printblock desde la barra lateral de herramientas.

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.

## Layout

- ✓ Printblocks para nuestro listado:



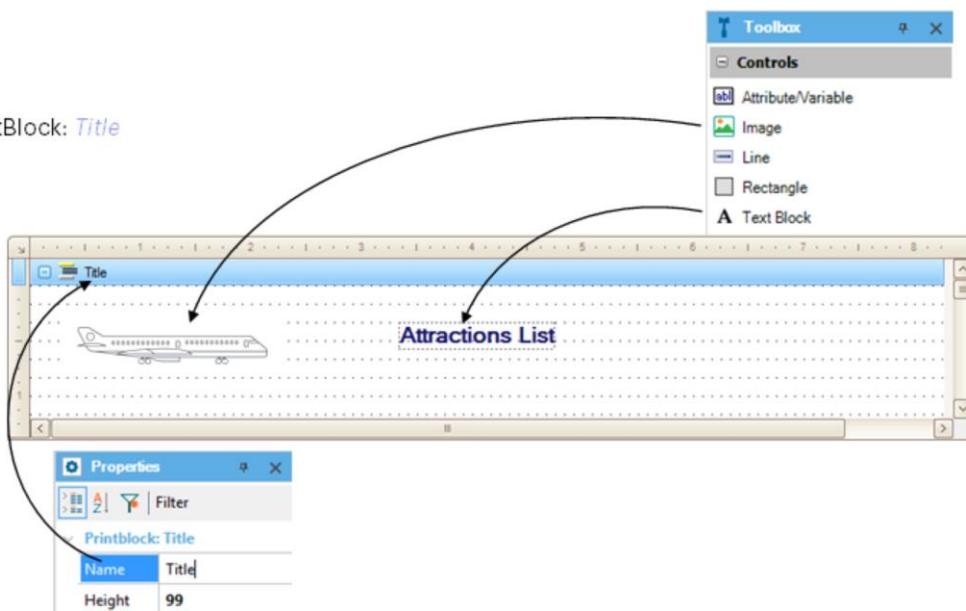
Podríamos definir entonces tres printblocks:

- uno en el cual incluiríamos el título del listado y la imagen, al que podríamos llamar Title
- otro para mostrar los títulos de las columnas con la línea debajo, al que llamaremos ColumnTitles
- y un tercer printblock en el cual vamos a desplegar los datos de las atracciones turísticas, al que llamaremos Attractions.

Empecemos entonces a definir esto.

## Layout

### ✓ 1. PrintBlock: Title



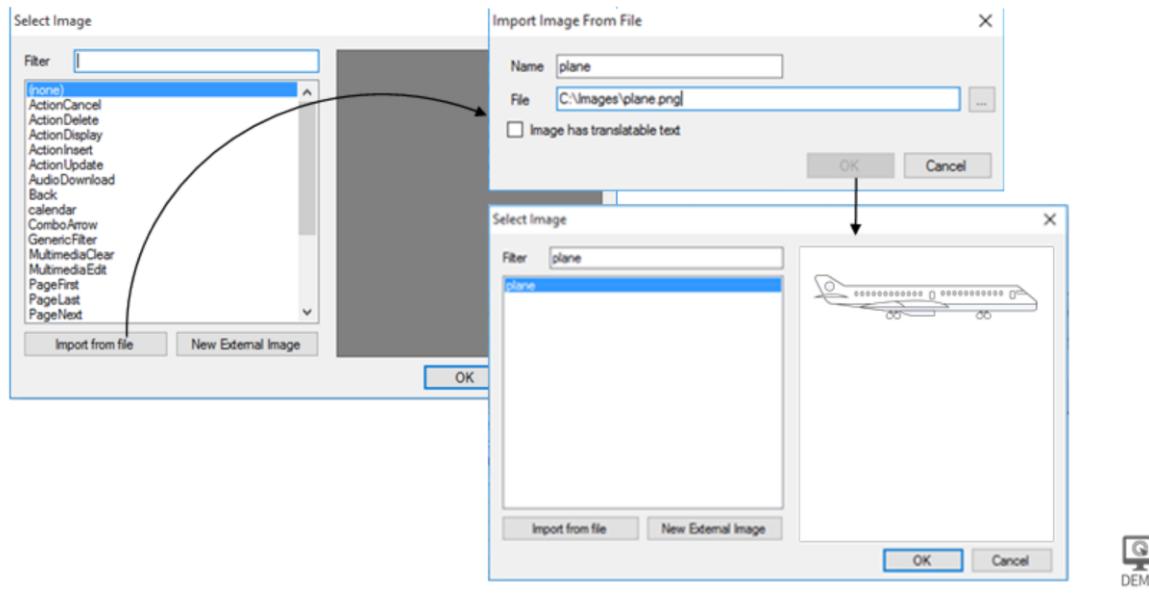
Podemos usar el printblock que se creó cuando creamos el objeto procedimiento para el título y la imagen.

Empecemos por el título. Para esto desde la Toolbox... arrastramos el control Text Block... editamos sus propiedades... y en la de nombre "Text" escribimos "Attractions List". Modificamos también su color, por MidnightBlue, su fuente... Size 14, Bold=True, 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.... Al hacerlo...

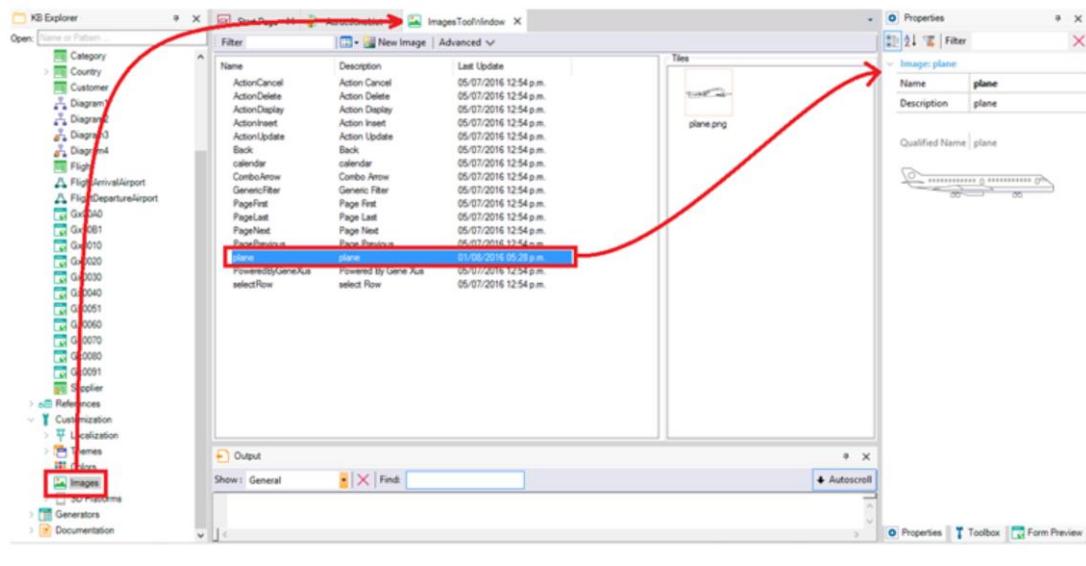
## Imágenes en la Base de Conocimiento



...se nos abre una 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.

El botón “Import from file” 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.

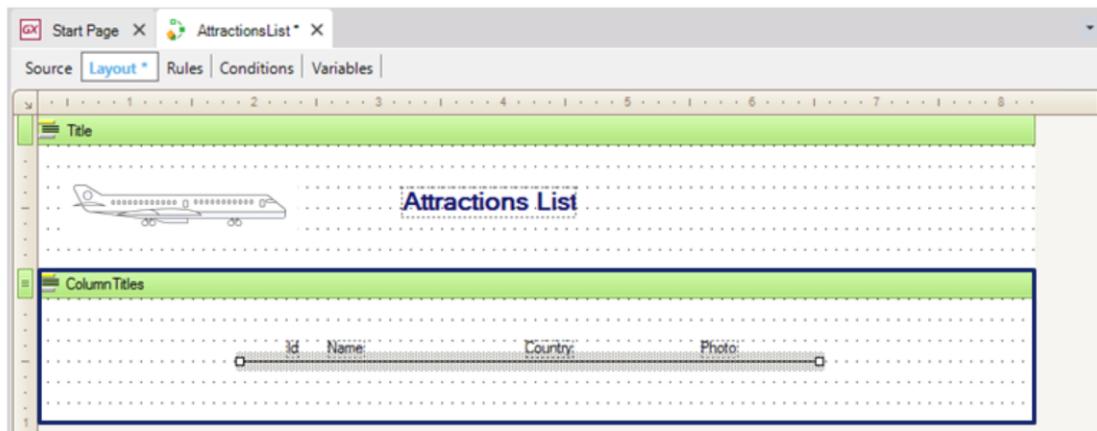
## Imágenes en la Base de Conocimiento



Observemos que desde KB Explorer/Customization/Images podemos acceder a todas las imágenes de la KB entre las que se encuentra la de nuestro avión.

## Layout

- ✓ 2. PrintBlock: *ColumnTitles*



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.

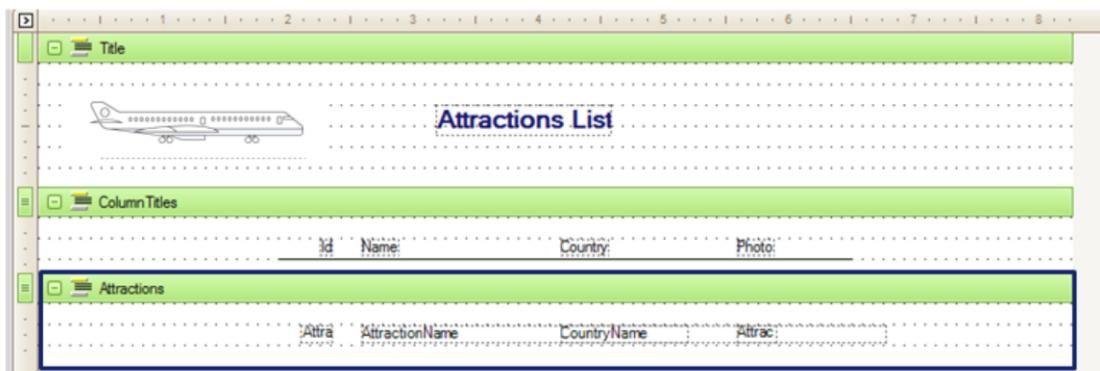
Así que desde la Toolbox, arrastramos el text block, y en su **propiedad Text** escribimos “Id”. Agregamos otro Textblock, y en su propiedad Text agregamos el texto “Name”. Y ahora otro Textblock para mostrar el texto “Country”. Y por último un text block para el título “Photo”.

Ubicamos los controles en la posición que queremos... Los podemos alienar seleccionándolos todos y luego: Menu/Layout/Align/Bottom.

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...

## Layout

- ✓ 3. PrintBlock: *Attractions*



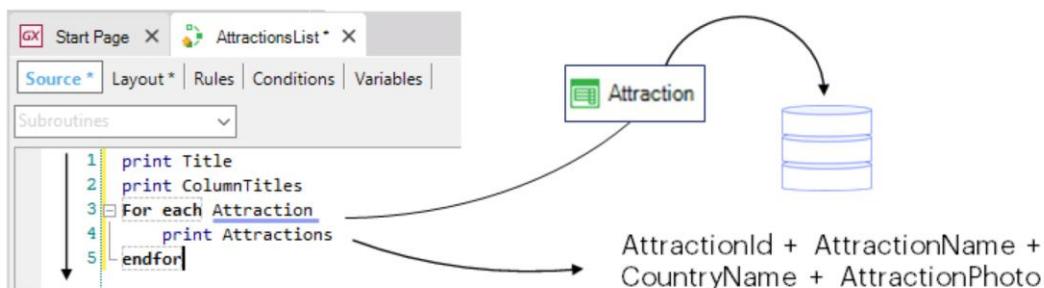
Nos está faltando ahora agregar el tercer printblock que habíamos dicho, para mostrar los datos de las atracciones turísticas, así que insertamos un nuevo printblock... y le ponemos como nombre: Attractions. Como los datos están almacenados en atributos, vamos nuevamente a la Toolbox, seleccionamos un control del tipo "Attribute/Variable" y lo arrastramos bajo el título "Id". En la ventana que se nos abre elegimos qué variable o atributo queremos mostrar en el control. Veamos que además de &today, en un procedimiento existen otras variables del sistema.

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

## Source

✓ Comandos:

- *Print* para imprimir un printblock y
- *For each* para recorrer una tabla y su extendida y hacer algo con cada registro de información.



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, **esta 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”...

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

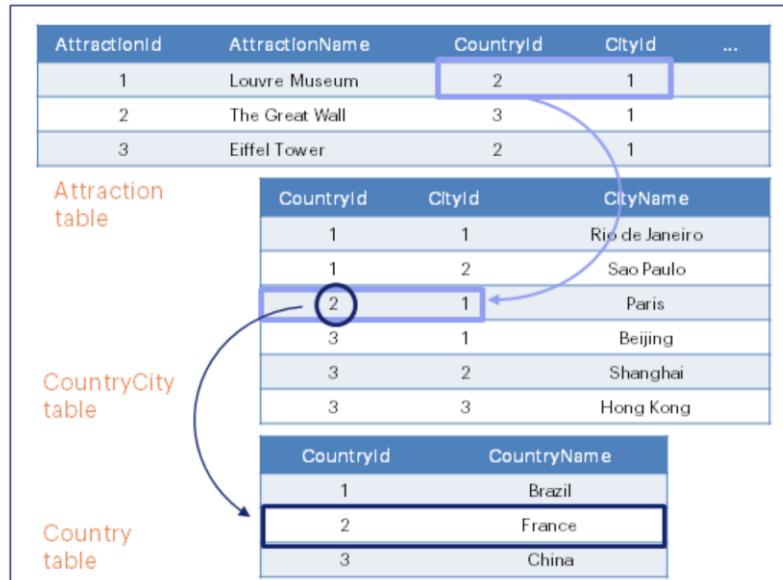
.. y ahora.. dado que queremos imprimir de cada atracción turística, el contenido de los atributos AttractionId, AttractionName, CountryName, y AttractionPhoto escribimos la orden para imprimir el printblock "Attractions" que los contiene. Así que escribimos Print Attractions.

Comando for each

For each

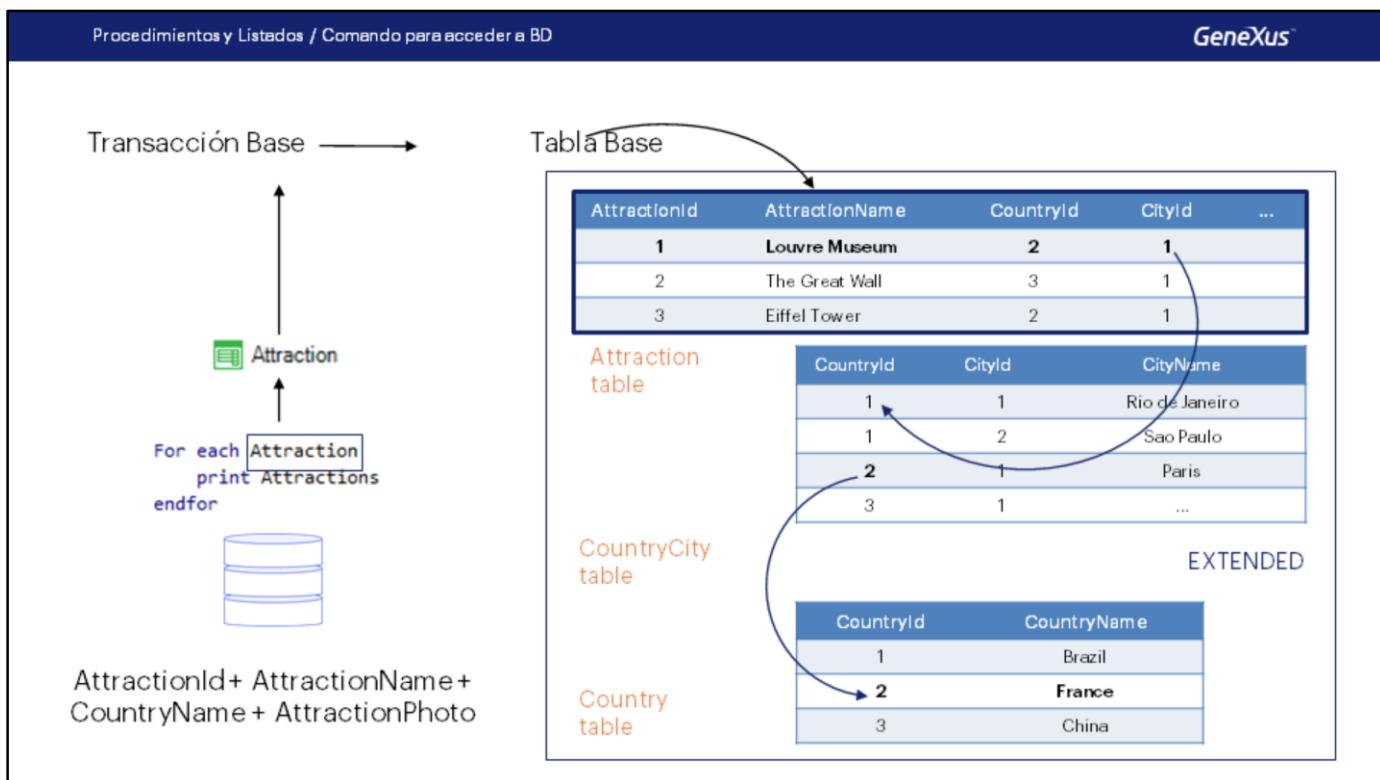


AttractionId+ AttractionName+  
CountryName + AttractionPhoto



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

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.

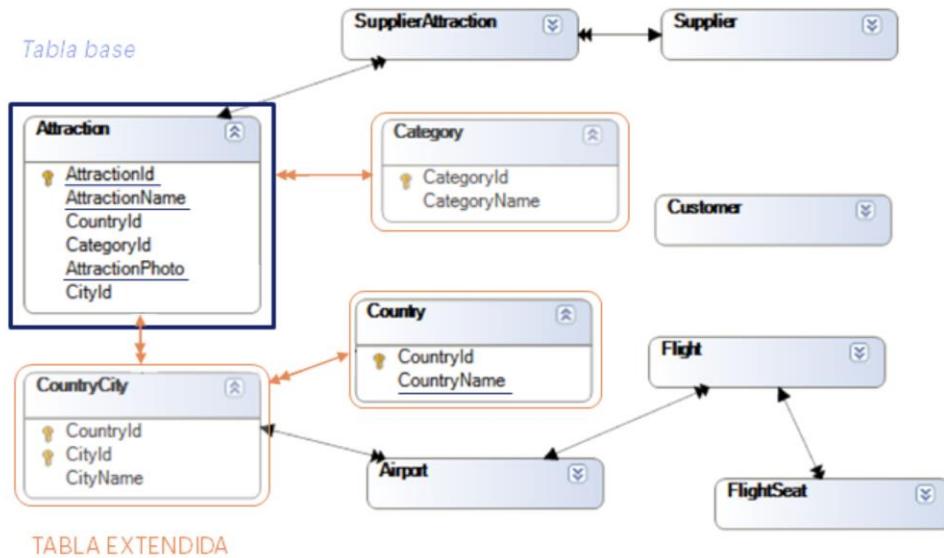


Vemos entonces que tenemos la **Transacción Base**, indicada al lado del For each. De la transacción base obtenemos la **Tabla Base**, que será la tabla que el for each navegará, recorriendo uno por uno sus registros (en este caso para imprimir un printblock).

Como en el printblock aparecen atributos no sólo de la tabla Attraction, sino de Country, para cada registro de la tabla Attraction en el que se esté posicionado en cada ejecución interna del for each, se accederá a un registro de CountryCity y desde él al correspondiente de Country, para recuperar el nombre del país de la atracción.

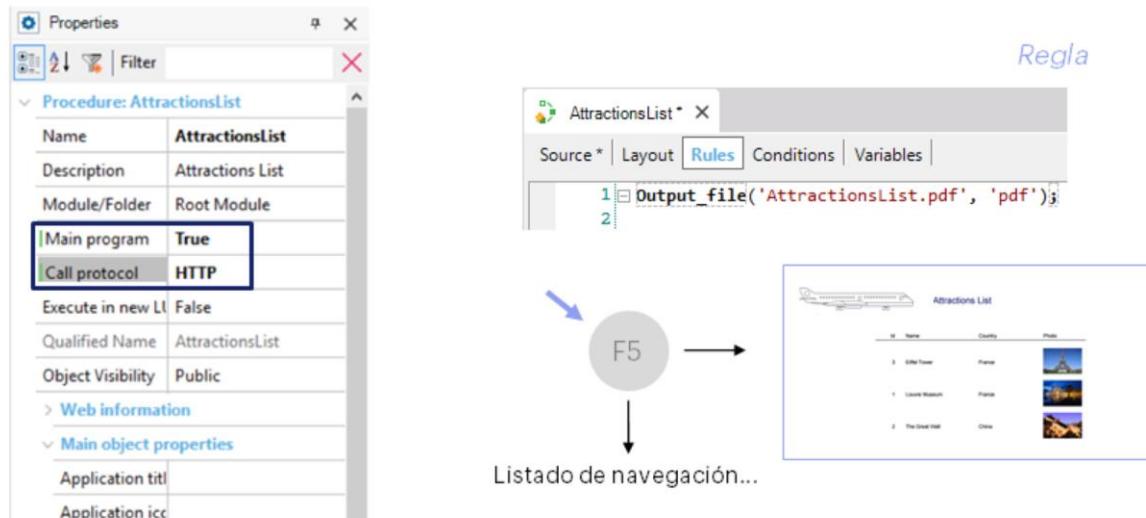
En definitiva, los atributos utilizados en el for each deberán pertenecer a **su tabla extendida**. Es decir, a la tabla extendida de su tabla base.

Transacción base, Tabla Base y Tabla Extendida de For each



Aquí vemos un diagrama con las relaciones entre las tablas de nuestra base de conocimiento. Dentro del for each tenemos los atributos AttractionId, AttractionName, AttractionPhoto y CountryName. Los primeros tres pertenecen a la tabla base del for each. Mientras que el último pertenece a una de las tablas de la extendida.

## Cómo generar listados en formato PDF: Propiedades del objeto Procedure



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. 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 Output\_file 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 veremos 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.

Además, en GeneXus se nos abre una ventana "Navigation View" con un reporte...

## Listado de navegación

The screenshot shows the GeneXus interface with the 'Navigation View' tab selected. The main content area displays the 'Procedure AttractionsList Navigation Report'. The report includes the following sections:

- Base Table:** Shows the procedure name 'AttractionsList' and its environment 'Default (C#)'.
- For Each Attraction (Line: 10):** Contains settings for the loop, including 'Order' (AttractionId), 'Index' (IATTRACTION), 'Navigation filters' (Start from: FirstRecord, Loop while: NotEndOfTable), and 'Join location' (Server).
- Join:** Details the joins between 'Attraction' (AttractionId) and 'Country' (CountryId).

Annotations numbered 1 through 4 highlight specific parts of the report:

- 1: Points to the 'Base Table' section.
- 2: Points to the 'For Each Attraction' loop.
- 3: Points to the 'Navigation filters' section.
- 4: Points to the joined tables 'Attraction' and 'Country'.

The status bar at the bottom indicates 0 Errors, 0 Warnings, and 1 Success.

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.

1. Podemos ver que al lado de donde dice "For Each", dice también Attraction, para indicarnos que esta 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.

2. También nos indica que para ordenar el listado de atracciones se utilizó el atributo AttractionId (que es la clave primaria de la tabla Attraction)...

3. 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. Mostró todas las atracciones....

4. 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.

## Cómo cambiar el orden de los datos

- ✓ Se pedía:  
listar en un archivo PDF todas las atracciones turísticas de la agencia de viajes, ordenadas alfabéticamente.

...se puede ordenar por cualquier atributo que pertenezca a la *tabla extendida* de la tabla que recorre el for each.

The screenshot shows the GeneXus Source editor interface. The top menu bar includes 'Source' (which is selected), 'Layout', 'Rules', 'Conditions', and 'Variables'. Below the menu is a dropdown menu labeled 'Subroutines'. The main area contains the following code:

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

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

Algo que quedó pendiente es listar las atracciones en orden alfabético, según el nombre de la atracción.

Podemos resolverlo simplemente agregando la cláusula “order AttractionName” luego de “For each Attraction”

## Listado de navegación

The screenshot shows the 'Navigation View' window in GeneXus. The title bar says 'Start Page X AttractionsList X Navigation View X'. The main area is titled 'Procedure AttractionList Navigation Report'. It contains several sections:

- Procedure Information:** Name: AttractionsList, Description: Attractions List, Output Devices: File, Main: Yes, Environment: Default (C#), Spec. Version: 15\_0\_0-104342, Form Class: Graphic, Program Name: AttractionsList, Call Protocol: HTTP, Parameters: None.
- Warnings:** A warning message: 'spc0038 There is no index for order AttractionName; poor performance may be noticed in group starting at line 3.'
- Levels:** A section titled 'For Each Attraction (Line: 10)' which includes:
  - Order: AttractionName (highlighted with a red box)
  - Navigation: Start from: FirstRecord
  - Filters: Loop while: NotEndOfTable
  - Join location: Server
- SQL Query:** The generated SQL code:

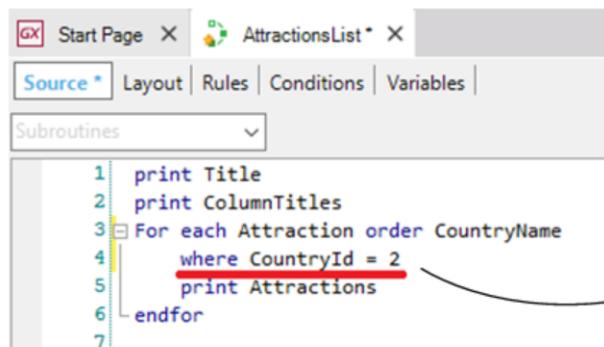
```
Attraction ('AttractionId') INTO CountryId AttractionPhoto.Uri  
AttractionPhoto AttractionId AttractionName  
Country ('CountryId') INTO CountryName
```
- Status Bar:** Shows 0 Errors, 0 Warnings, and 1 Success.

Por ahora no prestemos atención a la advertencia que el listado nos informa. Pero observemos cómo el listado de navegación nos informa por qué atributo se ordenará la Salida.

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.

## Cómo definir filtros

- ✓ Se desea: listar todas las atracciones turísticas de Francia.



```

GX Start Page X AttractionsList* X
Source * Layout | Rules | Conditions | Variables |
Subroutines ▾

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

```

The code shows a For Each loop for 'Attraction' ordered by 'CountryName'. The fourth line contains a red underline over 'CountryId = 2', indicating it is the part being explained. An arrow points from this line to the 'Listado de navegación...' text below.



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

where CountryName = "France"

Listado de navegación...

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

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.

En lugar de filtrar por el identificador de país, podríamos haber escrito **Where CountryName='France'**.

## Cómo definir filtros: listado de navegación

The screenshot shows the GeneXus Navigation View window for the AttractionsList procedure. The window has tabs at the top: Start Page, AttractionsList, and Navigation View. The AttractionsList tab is selected. The main area displays the following information:

**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 `CountryName`; poor performance may be noticed in group starting at line 3.

**Levels**

For Each Attraction (Line: 10)

Order: `CountryName` ←  
| No index

Navigation filters:  
Start from: `CountryName = 'France'`  
Loop while: `CountryName = 'France'`

Join location: Server  
↳=Attraction (AttractionId)  
↳=Country (CountryId)

At the bottom of the window, there are status indicators: 0 Errors, 1 Warning, and 0 Success.

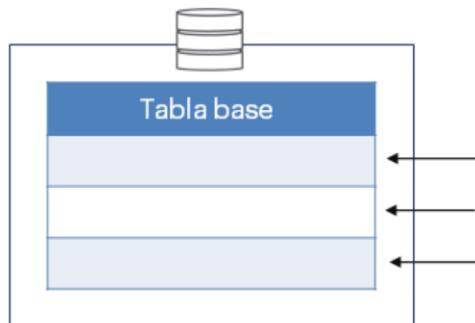
Vemos que ya no se recorre toda la tabla Attractions. Como estamos ordenando por CountryName, para quedarnos con los países de nombre 'France' solamente tiene que recorrer un pedacito de la tabla y no toda. Es análogo a buscar en un diccionario la palabra 'France'. No se busca en todo el diccionario.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

## Conceptualizing

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

Transacción Base  
For each *TransactionName.LevelName*

...



endfor

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 de la transacción, o más precisamente, el nombre del nivel de la transacción cuya tabla asociada queremos recorrer.

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

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

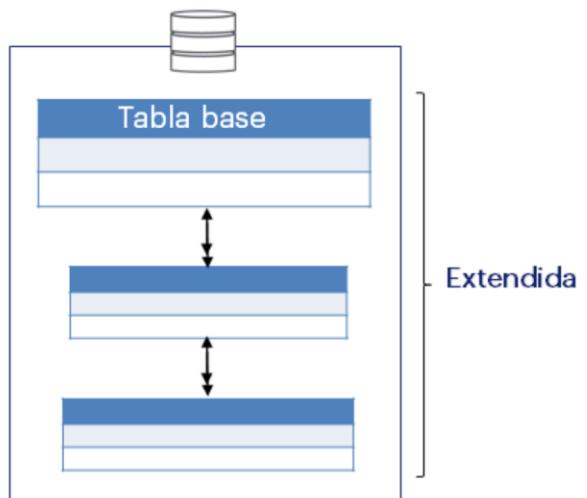
## Conceptualización

**For each *TransactionName.LevelName***

...

Los atributos aquí utilizados deben pertenecer a la *tabla extendida* de la tabla base a recorrer.

**endfor**



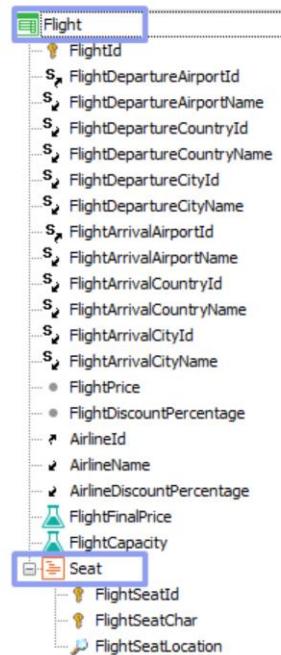
El conjunto de atributos que están entre el For each y el Endfor deben pertenecer a la tabla extendida de esa tabla base a recorrer.

## Conceptualizing

### ✓ Base Transaction:

- 1er nivel trn = **TransactionName**  
**For each Flight**  
**Where** FlightDepartureAirportId = 1  
...
**Endfor**

- Nivel anidado trn = **TransactionName.LevelName**  
**For each Flight.Seat**  
**Where** FlightId = 15  
...
**Endfor**



En el primer ejemplo estamos queriendo navegar la tabla de los vuelos que salen del aeropuerto 1. Para un primer nivel, el nombre de la transacción coincide con el del nivel.

En el segundo ejemplo estamos queriendo navegar los asientos del vuelo 15.

## Sintaxis del For each

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>*

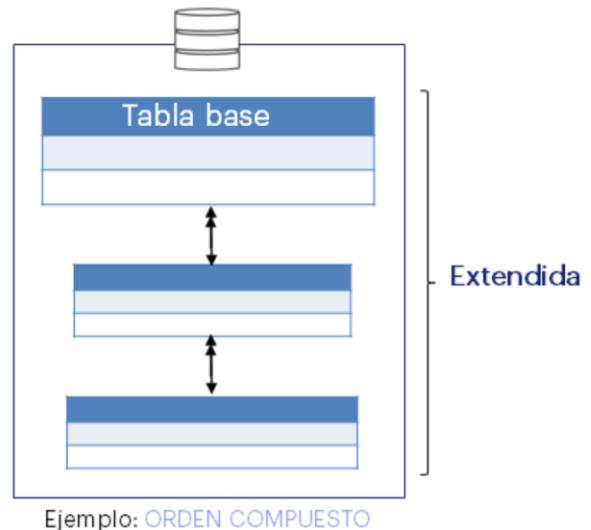
*MainCode*

    endfor

Aquí resumimos lo que hemos visto hasta el momento del comando for each. Iremos ampliando esta sintaxis a medida que vayamos viendo más temas.

## Sintaxis del For each

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



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.

## Ejemplo orden compuesto

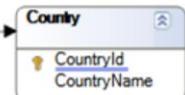
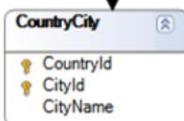
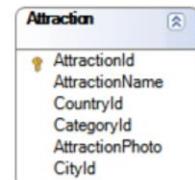
```
print Title
print ColumnTitles
For each Attraction order CountryName, CategoryName
    print Attractions
endfor
```

Attractions



Attractions List

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



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

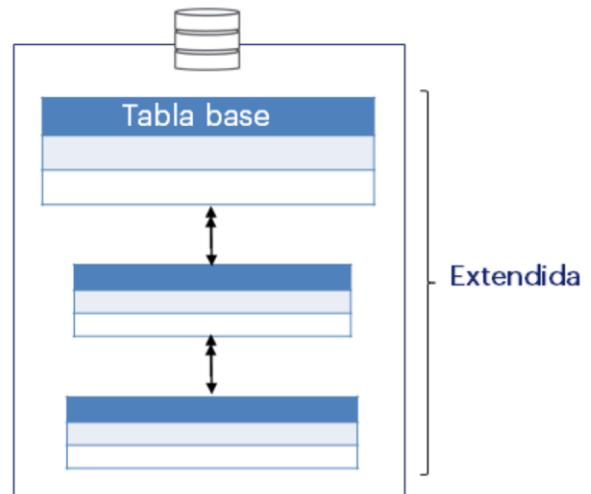
Aquí tanto CountryName, como CategoryName no están presentes en la tabla base, Attraction, sino en tablas de la extendida.

Sintaxis del For each: where

```

For each BaseTransaction
    order att1, att2, ..., attn
    where condition1
        conditiona and conditionb
        conditiona or conditionb
    MainCode
endfor

```



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.

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

- *Condition<sub>a</sub> AND Condition<sub>b</sub>*: lo que significa que ambas deben cumplirse a la vez
- *Condition<sub>a</sub> OR Condition<sub>b</sub>*: lo que significa que de cumplirse una de las dos, ya alcanza para que el registro que se está evaluando pase exitosamente el filtro:

Sintaxis del For each: where

**Foreach** *BaseTransaction*

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

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

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

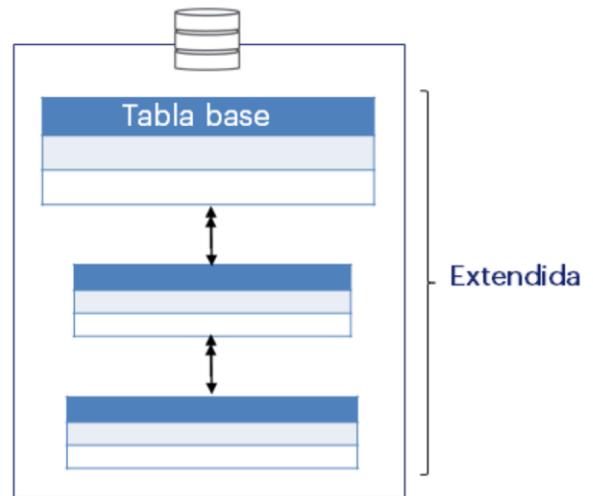
...

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

        and  
        and  
        and

**MainCode**

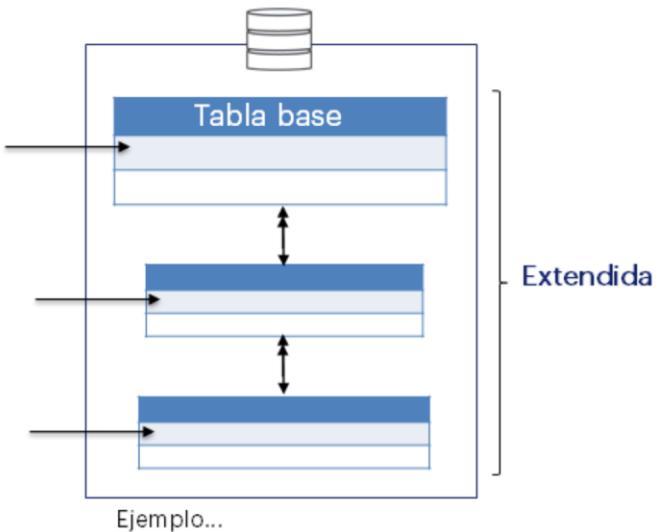
**endfor**



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

Sintaxis del For each: Main Code

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

## Sintaxis del For each: Main Code

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



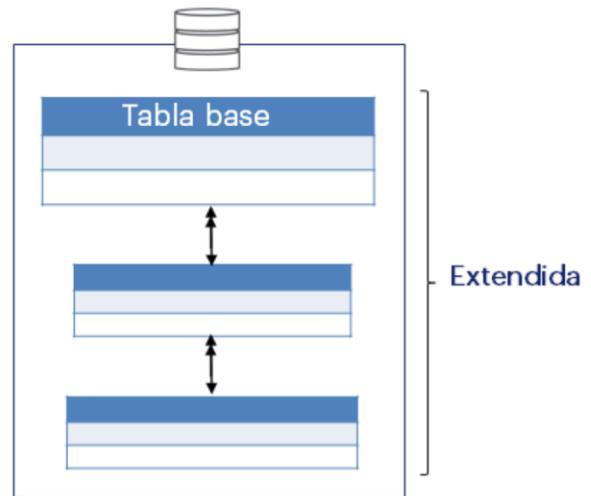
Attractions List

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

Por ejemplo, imprimir un printblock.

## Sintaxis del For each

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



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

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



Videos

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

Documentation

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

Certifications

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