

INLINE FORMULAS

GeneXus training
training.genexus.com

Anteriormente vimos fórmulas para definir un atributo de una transacción, veremos ahora cómo se pueden utilizar fórmulas en el código de un objeto.

Inline Formulas

The screenshot displays three examples of inline formulas in GeneXus code:

- Loop Condition:** In a `For each` loop, the condition `Count(AttractionName) > 10` is circled in red.
- Event Load Condition:** In an `Event Load` block, the condition `Count(ShowName) > 10` is circled in blue.
- Data Attribute Formula:** In a data attribute definition, the formula `Sum(ShowProductionAmount, ShowProductionPayment=Payment.Cash)` is circled in green.

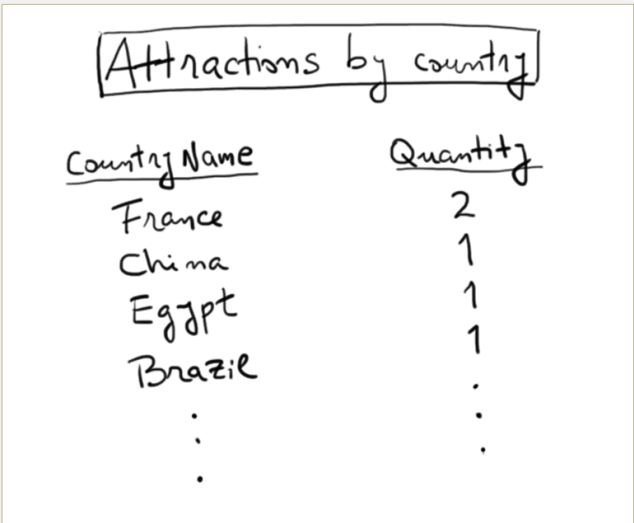
Inline formulas

GeneXus training

Una fórmula “inline” es una fórmula que declaramos como instrucción puntual dentro de determinado código, como en el source de un procedimiento, en un evento de un web panel, etc.

La misma solamente se conoce en el objeto donde fue definida.

Veamos algunos ejemplos de uso.

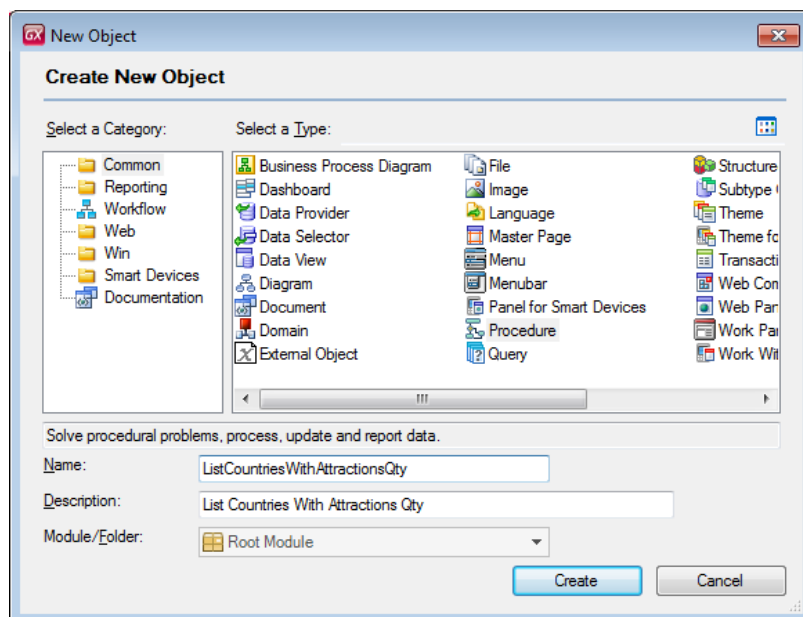


Country Name	Quantity
France	2
China	1
Egypt	1
Brazil	1
⋮	⋮

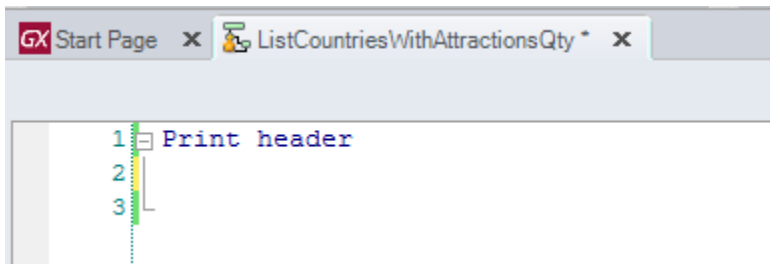
Inline formulas GeneXus training

Supongamos que la agencia de viajes nos solicita un listado que muestre todos **los nombres de países** y para cada país, **la cantidad de atracciones turísticas que ofrece**.

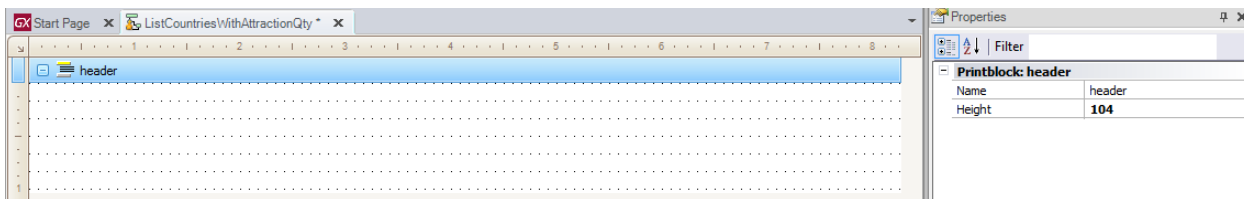
Para resolver esta solicitud, vamos a crear un objeto procedimiento y le damos el nombre: ListCountriesWithAttractionsQty.



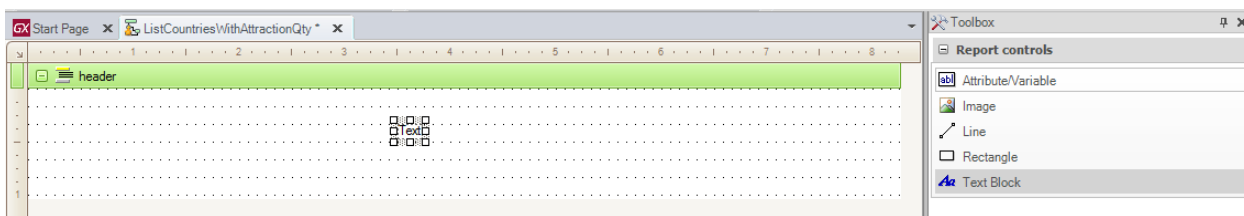
Quedamos posicionados en el source del procedimiento y lo primero que vamos a tener que hacer es imprimir un cabezal, así que escribimos la instrucción: Print header



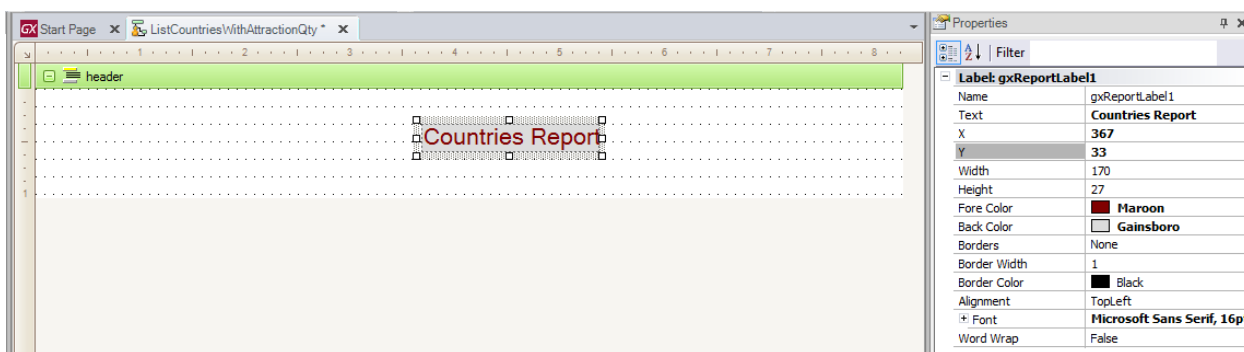
Vamos a la sección layout para definir el printblock de nombre header y al printblock que se nos ofrece por defecto, le asignamos dicho nombre



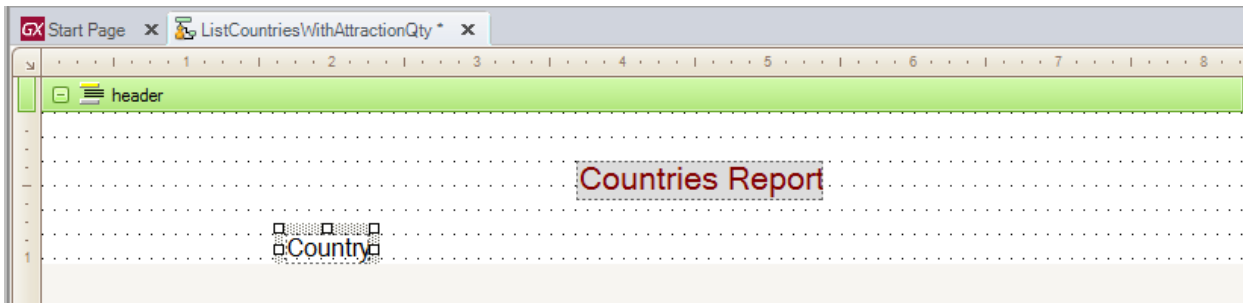
Arrastramos desde la toolbox un control Textblock al printblock:



Digitamos en la propiedad text del textblock: "Countries Report" y modificamos el tamaño de su fuente, el color de la fuente y el color de fondo:



Vamos a agregar al printblock otro texto: "Country" y le definimos un tamaño de fuente de: 12 .

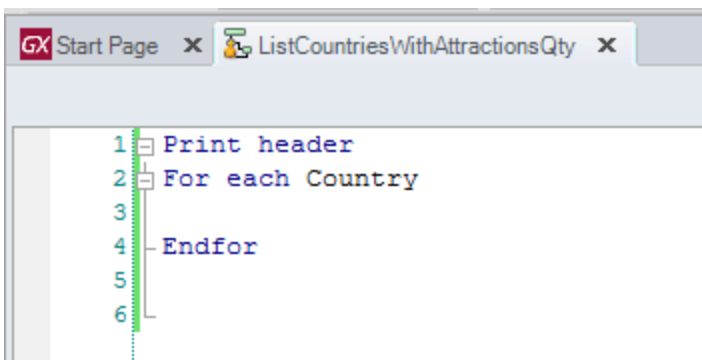


Copiamos este textblock y lo pegamos, para ganar tiempo. Solamente modificamos la propiedad text del nuevo textblock a: "Quantity"

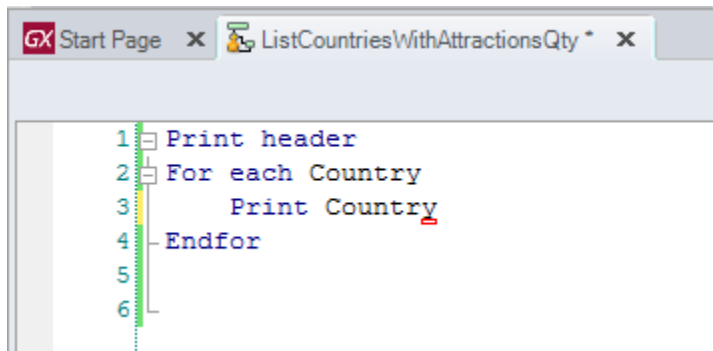


Volvamos al source para seguir implementando el reporte.

A continuación de la instrucción que imprime el cabezal, debemos escribir un comando For each, para acceder a la tabla que almacena los países y mostrarlos.

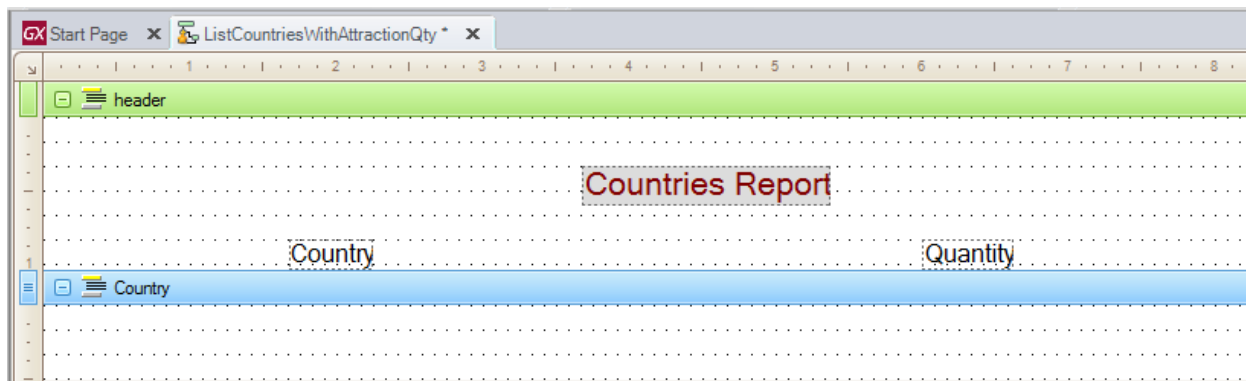


Dentro del For each escribimos: Print Country ...

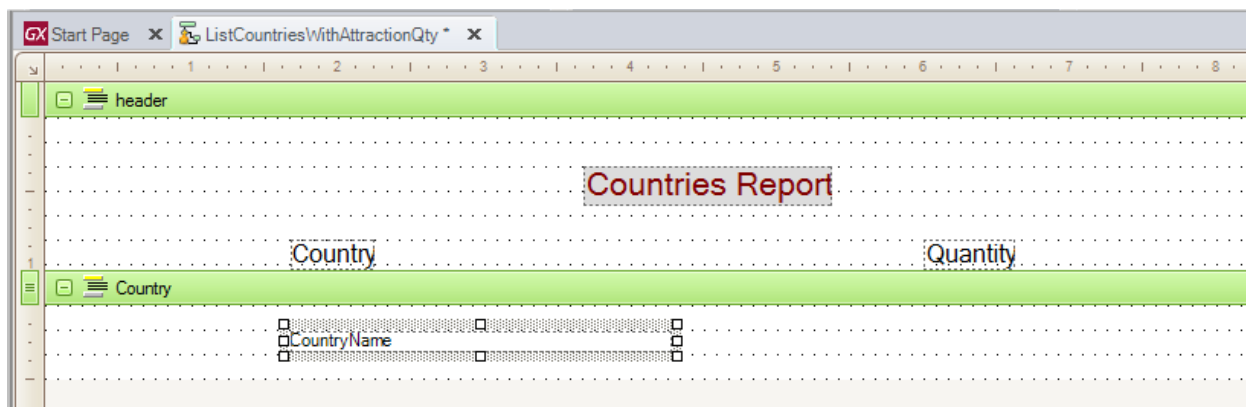


con el objetivo de mostrar en dicho printblock al atributo CountryName.

Vamos al layout e insertemos un nuevo printblock. Lo nombramos: Country y dentro agregamos al atributo CountryName.



Movamos al atributo para que los nombres de países se impriman debajo del título Country.



Volvamos al source.

Este For each así definido, navega toda la tabla de países y muestra el nombre de cada país consultado.

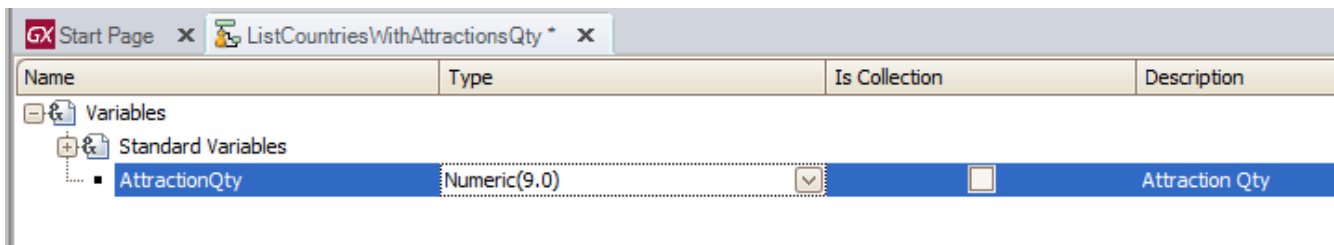


Solamente nos está faltando resolver el requisito de que para cada país, se imprima la cantidad de atracciones que tiene. ¿Cómo podremos implementar esto?

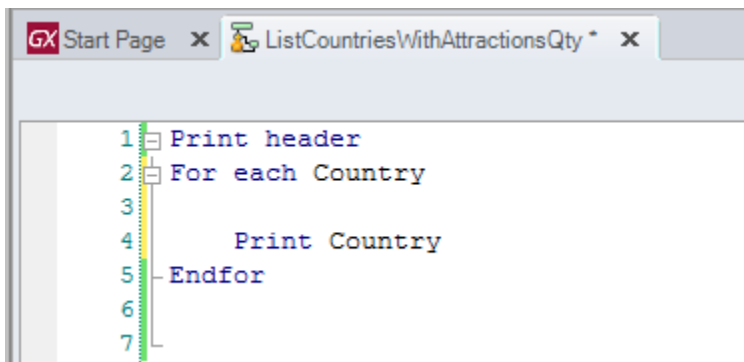
Calcularemos la cuenta con una fórmula “inline”... y mostraremos el resultado.

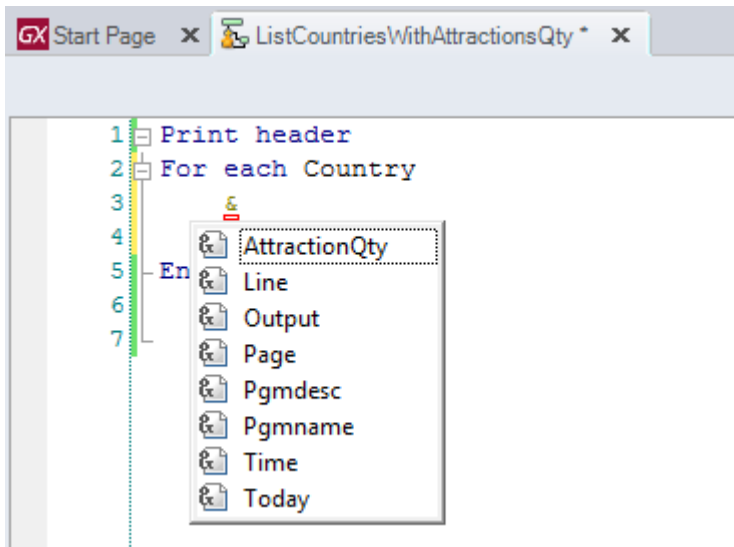
Vamos a definir una variable para asignarle a la misma el resultado que devuelva una fórmula Count y luego mostraremos la variable en el printblock.

La creamos, con nombre AttractionQty, de tipo Numeric(4) :

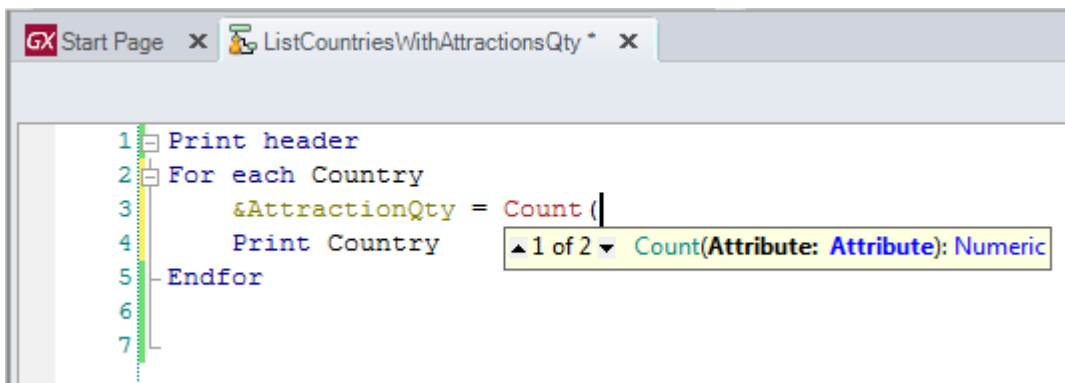


Volvamos al source y dentro del For each, justo antes de imprimir, incluimos a la variable: &AttractionQty...

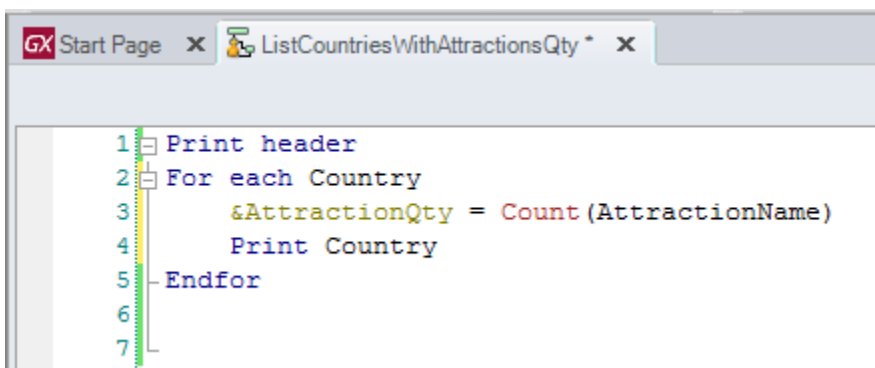




a la cual le asignamos el resultado que devuelva una fórmula Count.



Dado que queremos contar atracciones, dentro del paréntesis debemos incluir un atributo de la tabla ATTRACTION. Escribimos...AttractionName.



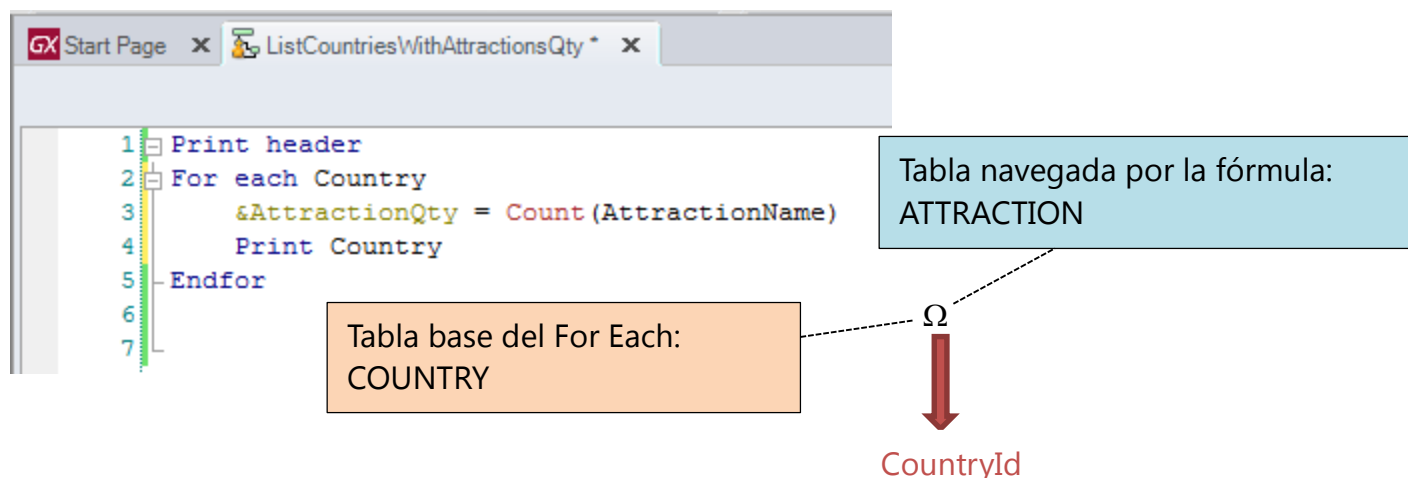
Vamos a analizar este código para ver cómo funciona.

Las fórmulas, determinan la tabla a navegar, por el o los atributos referenciados dentro del paréntesis.

En este caso hemos incluido al atributo AttractionName, porque pretendíamos contar atracciones y GeneXus lo entiende exactamente así.

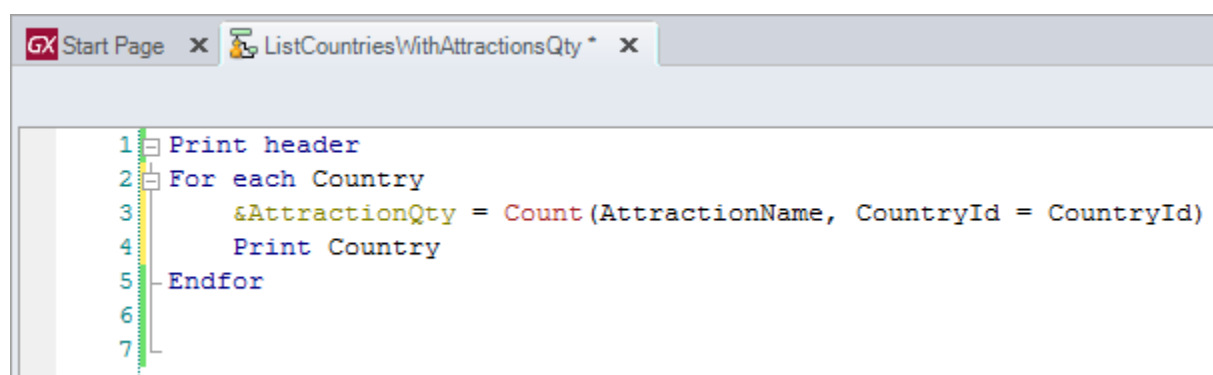
Si la fórmula no estuviera dentro de un For each, ni en un contexto en el cual ya se está recorriendo una tabla, **se contarían todas las atracciones**.

Pero como la fórmula está definida dentro de un comando For each, luego de determinada la **tabla que navega la fórmula**, GeneXus analiza si encuentra relación entre dicha tabla y **la tabla base del For each**.



Vemos que en este caso, hay un atributo en común entre las 2 navegaciones (CountryId), **así que para cada país que el For each encuentre**, se cuentan sus atracciones

Es como definir esta condición de filtro **explícita en la fórmula**, pero no es necesario escribirla porque GeneXus aplica este filtro automáticamente al contar.



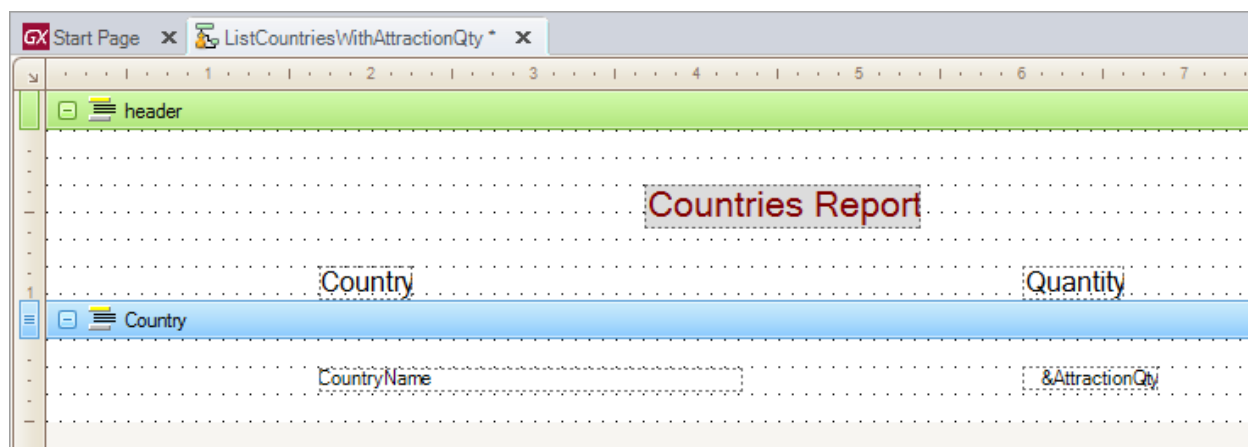
Ahora, solamente nos resta agregar la variable &AttractionQty al printblock Country, para que inmediatamente después de tener la cuenta en la variable, mostremos al lado del nombre del país, su cantidad de atracciones.

```

GX Start Page x ListCountriesWithAttractionsQty * x
1 Print header
2 For each Country
3     &AttractionQty = Count(AttractionName)
4     Print Country
5 Endfor
6
7

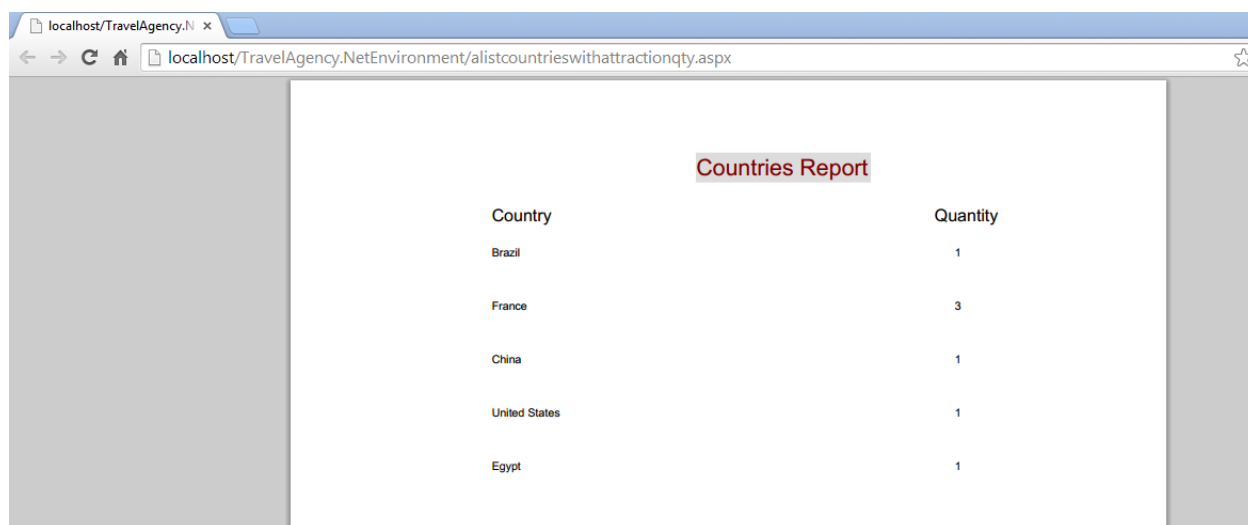
```

La agregamos.... y la alineamos con respecto a su título:



No debemos olvidar configurar las propiedades del reporte, para emitirlo en ambiente web y formato PDF

Grabamos y lo ejecutamos.



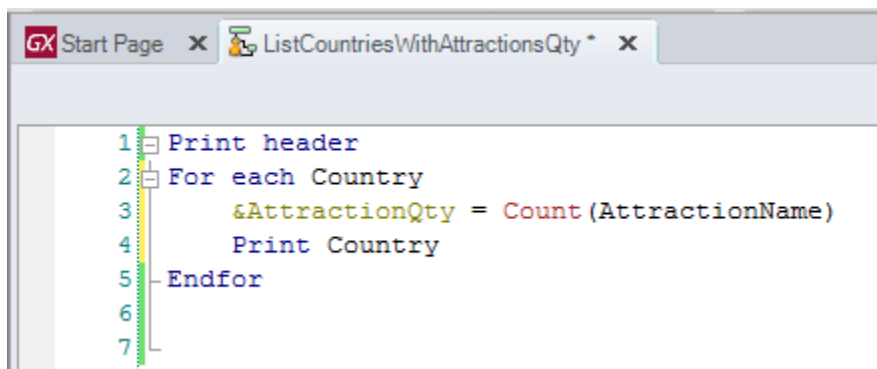
De esta manera hemos resuelto el requisito que nos solicitaron.

Como hemos visto, para determinar la tabla a ser navegada por la fórmula, no se tuvieron en cuenta los atributos referenciados en el For each sino solamente los incluidos en la definición de la fórmula.

Y de igual manera, para determinar la tabla a ser navegada por el For each, no se tuvieron en cuenta los atributos referenciados en la fórmula, sino solamente los usados en el cuerpo del For each.

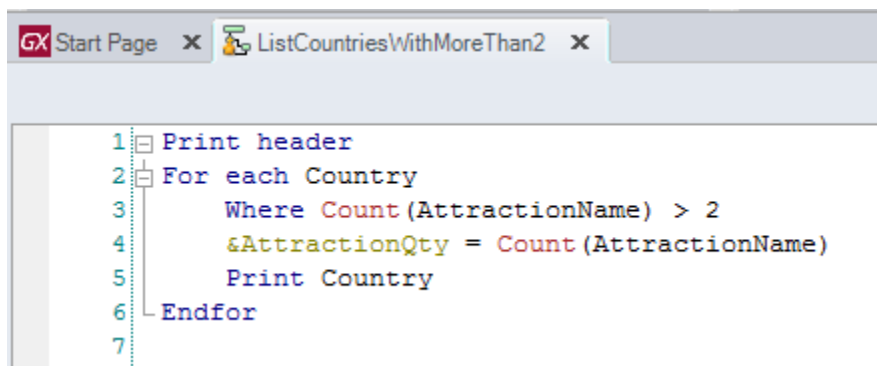
Muy bien... Supongamos que ahora nos piden que listemos todos los países que tienen **más de 2 atracciones para visitar**.

Salvemos este procedimiento con otro nombre y vamos a hacerle unas pocas modificaciones al source...



```
1 Print header
2 For each Country
3     &AttractionQty = Count(AttractionName)
4     Print Country
5 Endfor
6
7
```

Dado que la condición que nos solicitaron que se cumpla para mostrar los países, es que tengan más de dos atracciones turísticas, agregaremos una cláusula Where al For Each... Pero en lugar de asignar el resultado de la fórmula Count a una variable y filtrar por ella, podemos resolver el requisito así:



```
1 Print header
2 For each Country
3     Where Count(AttractionName) > 2
4     &AttractionQty = Count(AttractionName)
5     Print Country
6 Endfor
7
```

Dado que este procedimiento fue grabado a partir de otro, tiene configurado todo lo necesario para ser impreso como PDF, así que lo ejecutamos y vemos el resultado.

The screenshot shows a web browser window with two tabs. The active tab is titled 'localhost/TravelAgency.N...' and the address bar shows 'localhost/TravelAgency.NetEnvironment/alistcountrieswithmorethan2.aspx'. The main content area displays a table titled 'Countries Report' in red text. The table has two columns: 'Country' and 'Quantity'. There is one data row showing 'France' under the 'Country' column and '3' under the 'Quantity' column.

Country	Quantity
France	3

Con esto completamos 2 ejemplos de uso de fórmulas “inline”, para obtener cálculos fácilmente.

Para finalizar, grabemos los cambios en Gxserver.