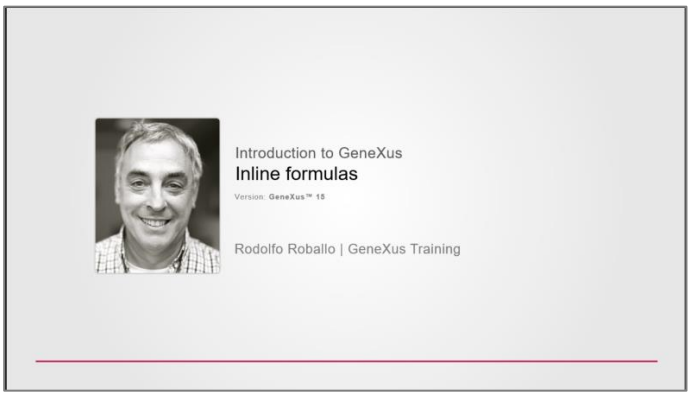
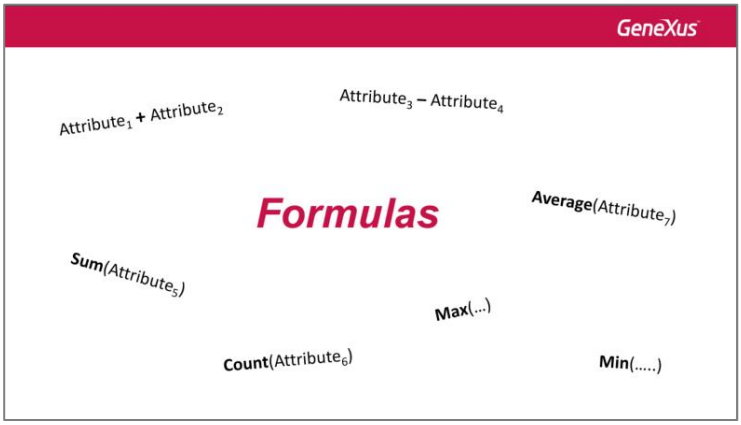
Fórmulas inline: fórmulas dentro del código

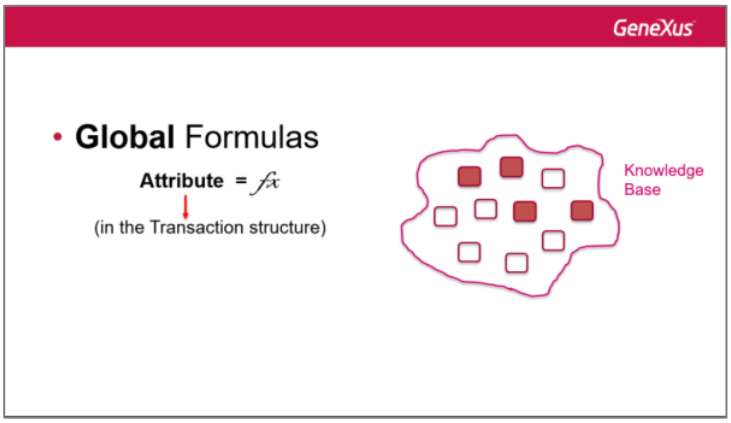


En una clase anterior ya habíamos distinguido entre fórmulas globales y fórmulas locales o inline.

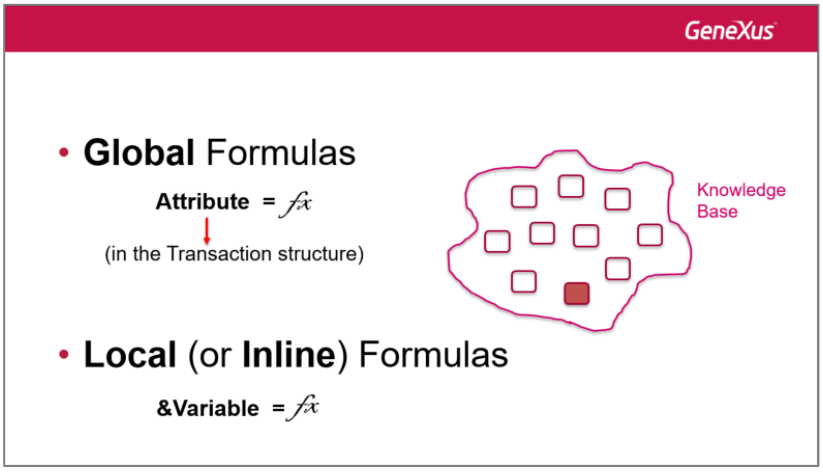
Las fórmulas **globales**



se definían a nivel de atributos en la estructura de una transacción, es decir, se especificaba que tal atributo siempre se calculaba de tal forma, y eso hacía que luego, en cualquier objeto en el que se necesitara recuperar el valor del atributo, esta fórmula se evaluaba para dar su resultado. Por eso los atributos fórmula eran atributos virtuales, que no se almacenaban en las tablas.



En cambio las fórmulas también se pueden utilizar en forma **local**, es decir, especificadas para evaluarse solamente en el lugar del código del objeto en el que se encuentren. Por ejemplo, en el Source de un procedimiento. Podríamos, por ejemplo, asignarle a una variable el resultado de una fórmula.



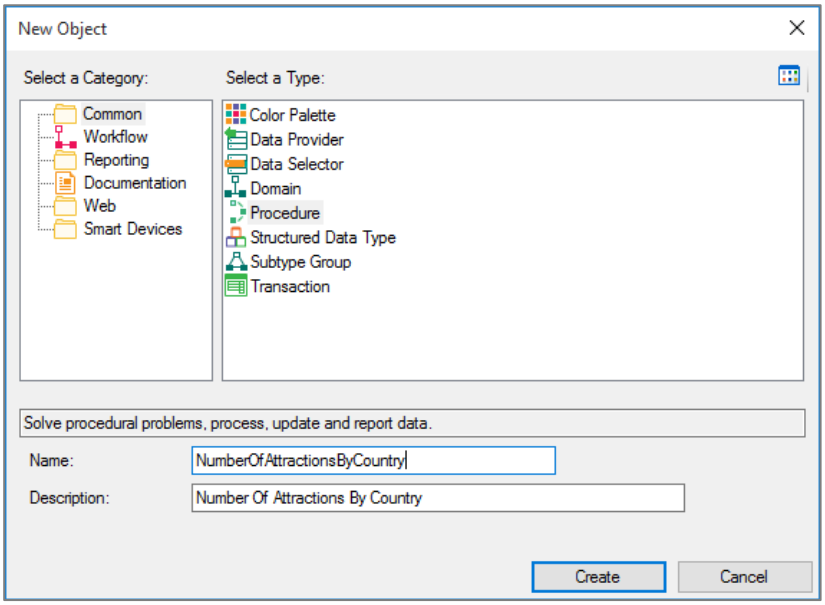
Veamos algunos ejemplos de uso.

Countries List

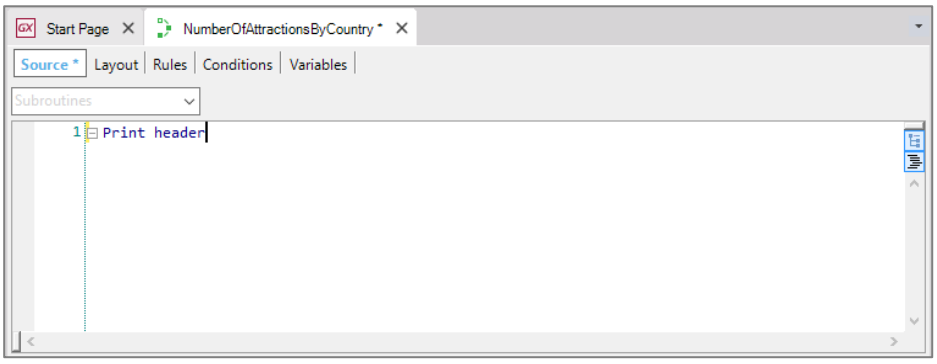
Country	Quantity
Brazil	1
France	3
China	2
United States	1

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**.

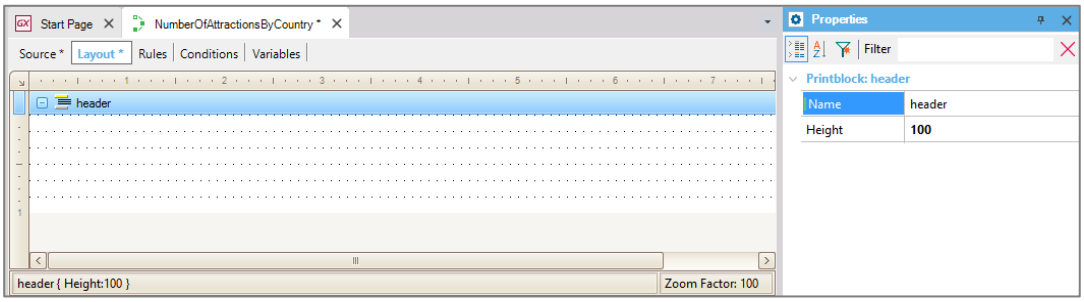
Para resolver esta solicitud, vamos a crear un objeto procedimiento:



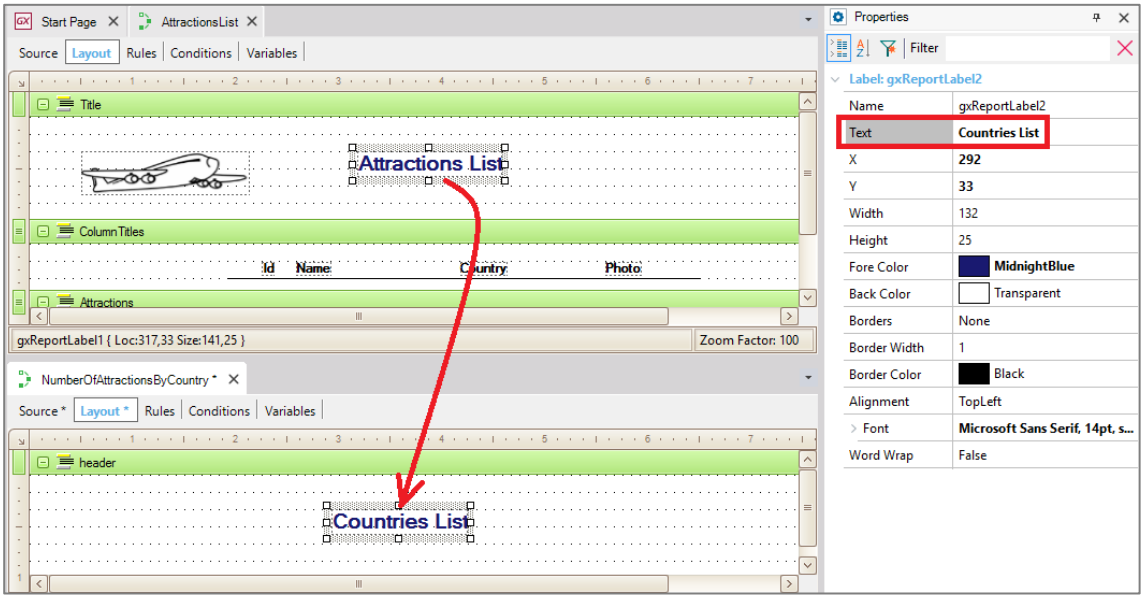
Quedamos posicionados en el Source del procedimiento y lo primero que vamos a hacer es imprimir un cabezal con un título, así que escribimos la instrucción: Print header.
Header será un printblock que todavía no tenemos. Entonces definámoslo.



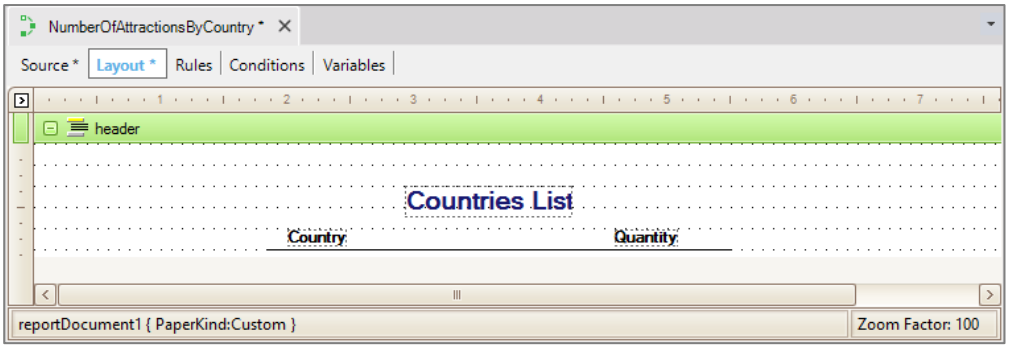
Vamos a la sección Layout y al printblock que tenemos por defecto le cambiamos el nombre por “header”:



Podemos ahora insertar de cero un textblock y darle el formato deseado, o ir al listado de atracciones que teníamos antes, y hacer copy&paste del textblock para conservar su formato, y cambiarle solamente la propiedad Text:

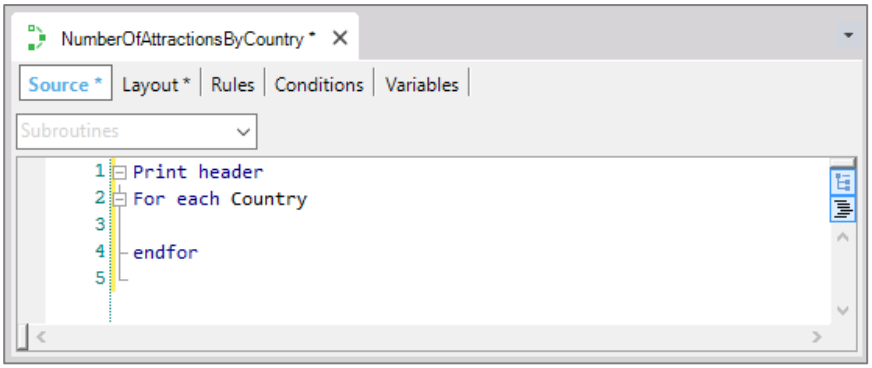


Aquí vemos que ya hemos agregado al printblock dos textos para las columnas del listado y una línea:

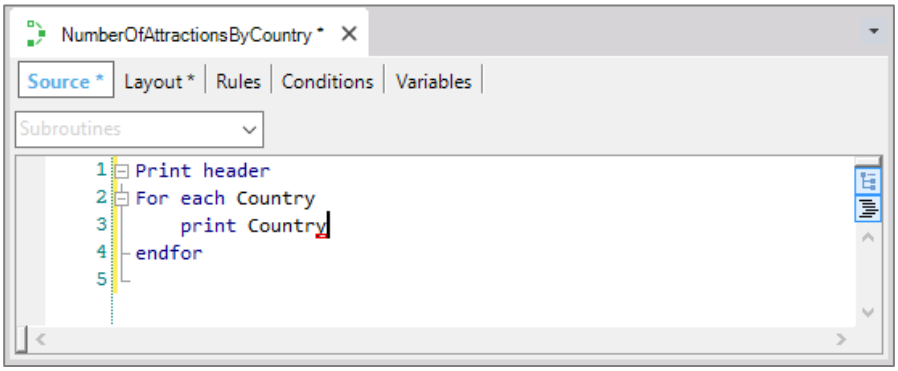


Volvamos al Source para seguir implementando el reporte.

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.

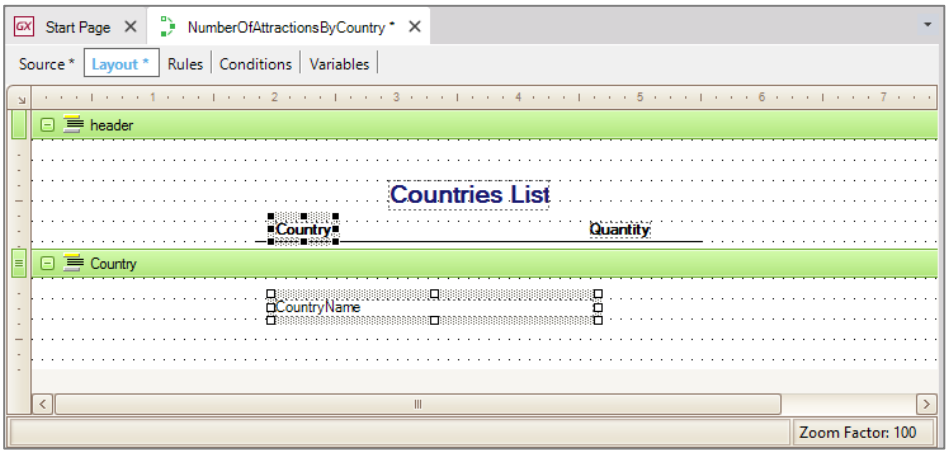


Dentro del For each escribimos: Print Country…



con el objetivo de mostrar en un printblock de ese nombre el atributo CountryName.

Por lo que tenemos que crear ese printblock.
…y dentro agregamos el atributo CountryName… alineándolo con la columna de título Country:



Volvamos al Source.

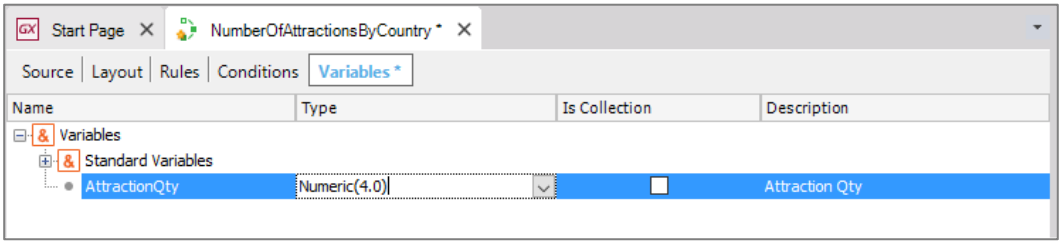
Este For each así definido navega toda la tabla de países y muestra el nombre de cada país consultado.

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 cantidad mediante una fórmula "inline"… y mostraremos el resultado.

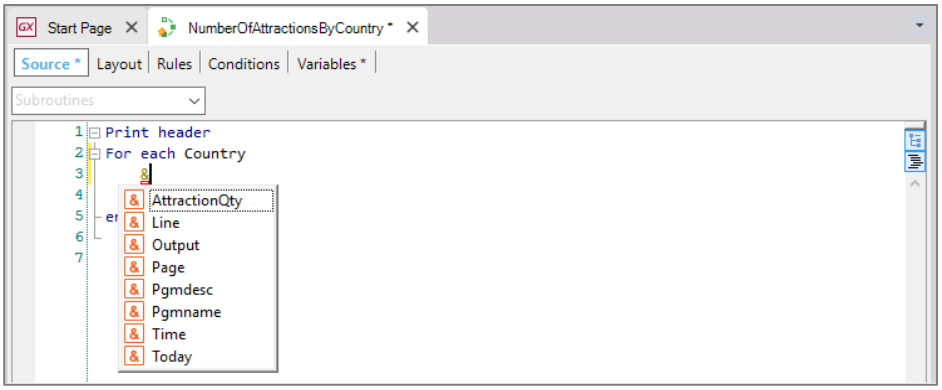
Vamos a definir una variable para asignarle el resultado que devuelva una fórmula Count y luego mostraremos la variable en el printblock.

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

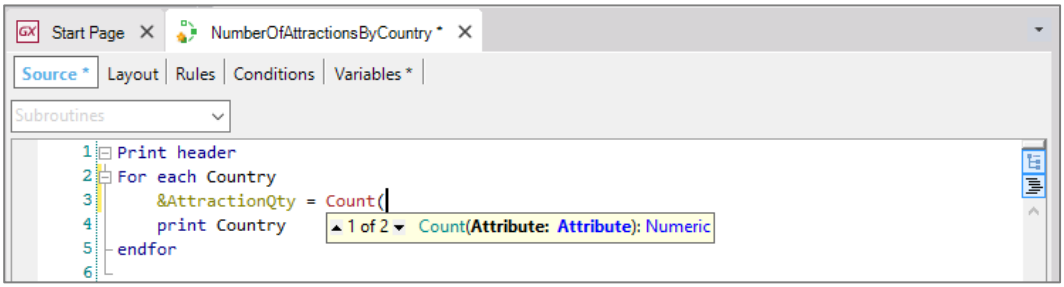


Recordemos que las variables son espacios de memoria locales al objeto donde se definen. Esto quiere decir que sólo existen durante cada ejecución de ese objeto. Cuando la ejecución termina, desaparecen.

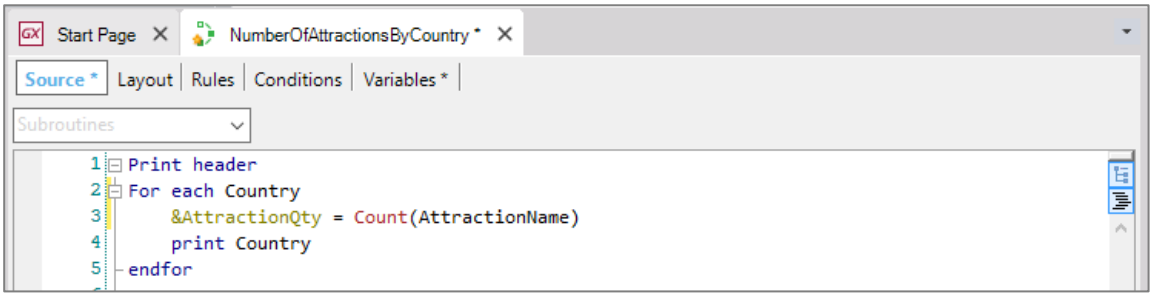
Volvamos al Source y dentro del For each, justo antes de imprimir, a la variable: &AttractionQty…



le asignamos el resultado de una fórmula Count… que va a contar las atracciones.



Dentro del paréntesis debemos incluir un atributo que haga que GeneXus sepa de qué tabla es que queremos que cuente. Como es de ATTRACTION, elegiremos algún atributo que sepamos que está en esa tabla. Por ejemplo, AttractionName.



Analicemos este código para ver cómo funciona.

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

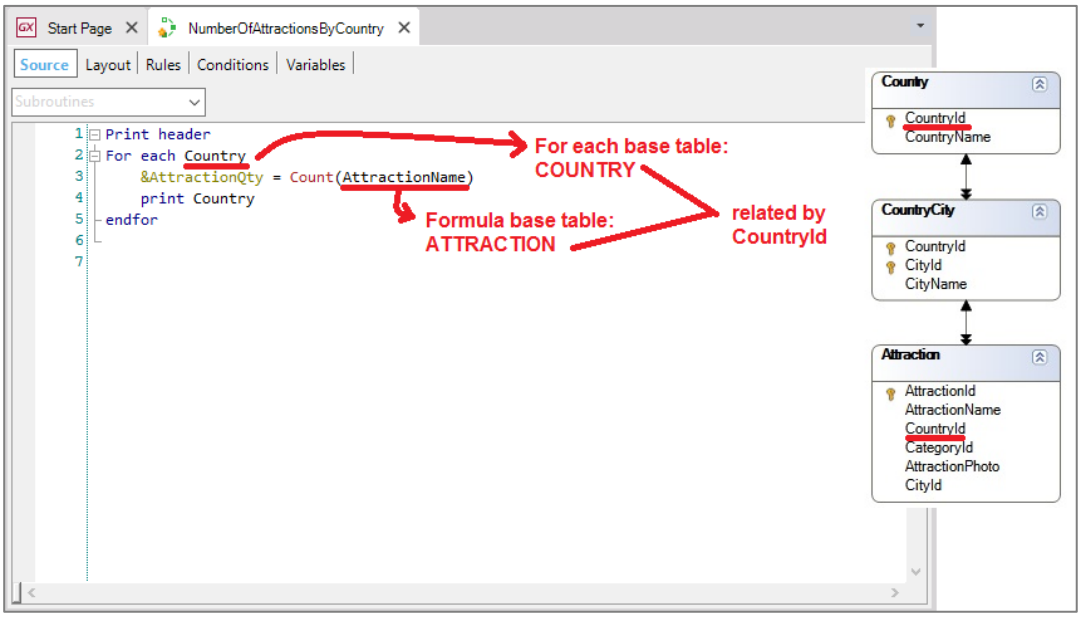
En este caso hemos incluido el atributo AttractionName, porque queremos contar atracciones y GeneXus lo

entiende exactamente así. Pero las atracciones que queremos contar no son todas las de la tabla, sino las que corresponden al país en el que estamos posicionados en cada ejecución del For each.

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

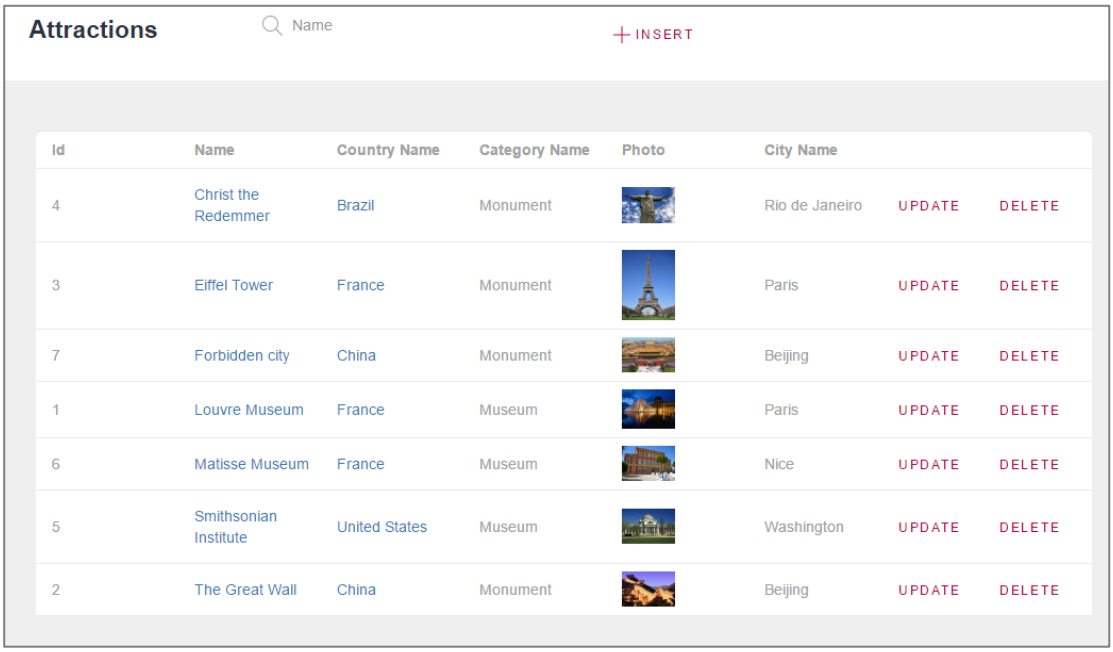
Pero como la fórmula está definida dentro de un comando For each es decir, se encuentra en un contexto en el que se está recorriendo una tabla, en este caso, la de países, será influenciada por ese contexto. Así, no se contarán TODAS las atracciones de la tabla, sino las relacionadas con el país que se está procesando en cada iteración del for each.

Dicho de otro modo: GeneXus determina la **tabla que debe navegar la fórmula** en base a sus atributos, y luego analiza si hay relación entre esa tabla y **la tabla base del For each (incluyendo a su extendida)**.



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

Así, si hemos agregado a las atracciones que ya teníamos el Museo Matisse de Francia y la ciudad prohibida de China:



El for each se ejecuta, y para el primer país, Brazil, se cuentan las atracciones con ese mismo CountryId…

Luego el for each pasa al siguiente registro, 2- France, y cuenta sus atracciones, las del CountryId = 2… Encuentra tres.

Luego itera el siguiente registro, 3 – China, contando 2 atracciones

Y por último United States, con una sola atracción.

GeneXus

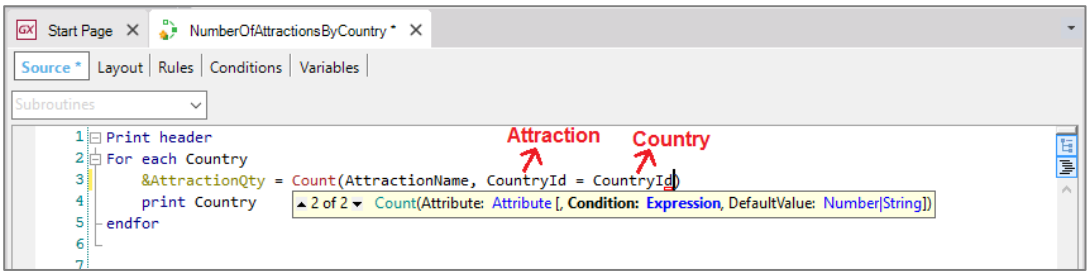
CountryId	CountyName
1	Brazil
2	France
3	China
4	United States

CountryId	CityId	CityName
1	1	Rio de Janeiro
1	2	Sao Paulo
2	1	Paris
2	2	Nice
3	1	Beijing
3	2	Shanghai
3	3	Hong Kong
4	1	New York
4	2	Washington

AttractionId	AttractionName	CountryId	CityId	...
1	Louvre Museum	2	1	
2	The Great Wall	3	1	
3	Eiffel Tower	2	1	
4	Christ the Redemmer	1	1	
5	Smithsonian Institute	4	2	
6	Matisse Museum	2	2	
7	Forbidden city	3	1	

Country	Quantity
Brazil	1
France	3
China	2
United States	1

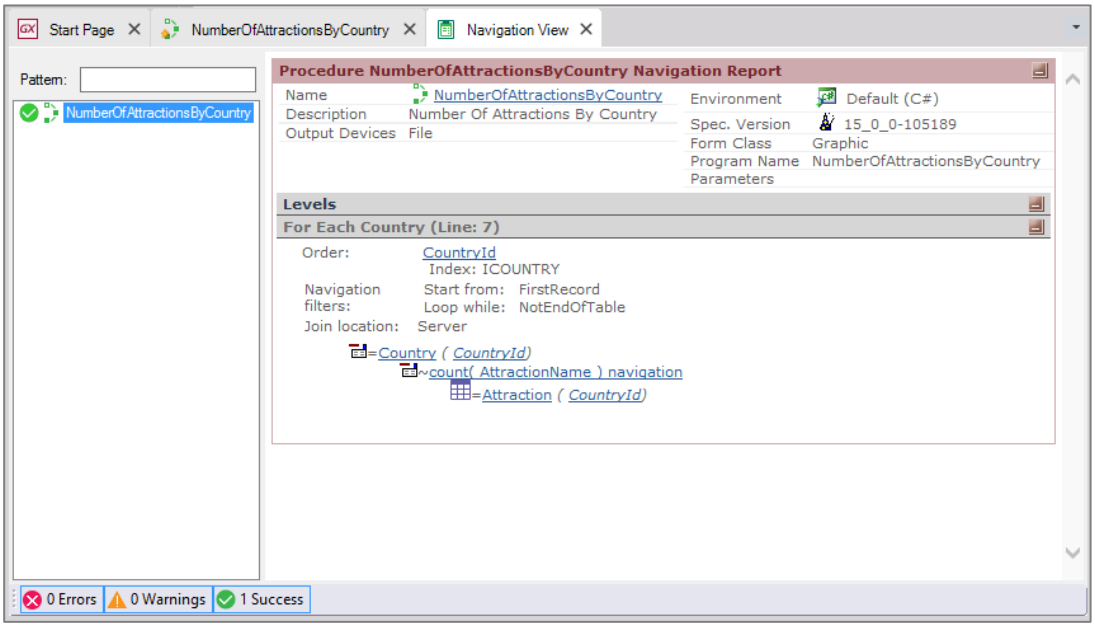
En definitiva, funciona como si hubiésemos definido esta condición de filtro **explícita en la fórmula**:



Donde un CountryId es el de la tabla navegada por la fórmula y el otro es el de la tabla del for each. No es necesario escribirla. GeneXus la detecta automáticamente.

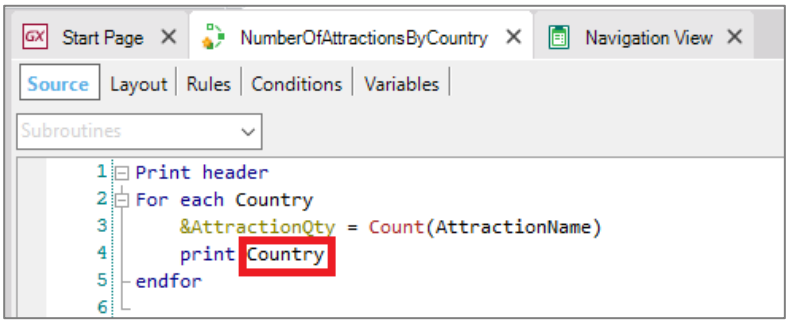
Otra cosa que detectará automáticamente es que le convendrá recorrer la tabla a ser navegada por la fórmula **ordenándola** por ese atributo de filtro. De lo contrario, como vemos en este ejemplo, para cada país tendrá que recorrer toda la tabla de atracciones para contar las que correspondan, cada vez.

Podemos ver todas estas inferencias que realiza GeneXus a través del listado de navegación.

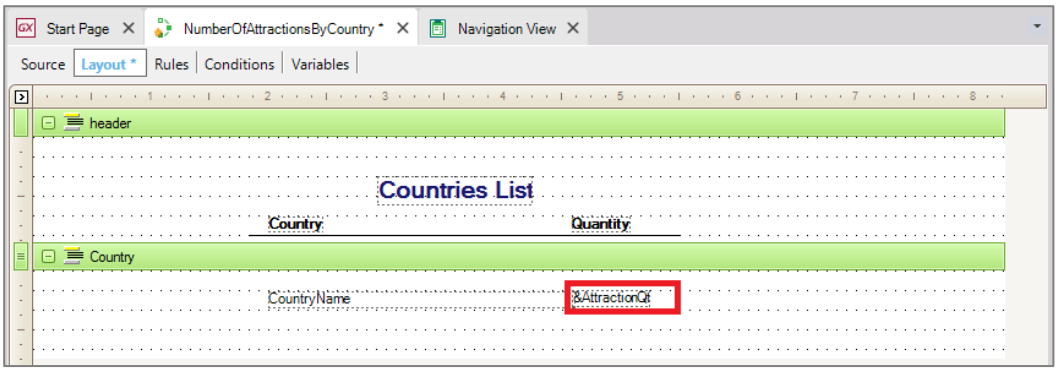


Vemos que nos muestra que el For each recorrerá la tabla Country, y que para cada país, realizará el count, para el que deberá navegar la tabla Attraction, por CountryId.

Solamente nos resta agregar la variable &AttractionQty calculada como primera instrucción en el cuerpo del For each, al printblock Country, para que inmediatamente después de tener la cuenta en la variable, se muestre al lado del nombre de país:

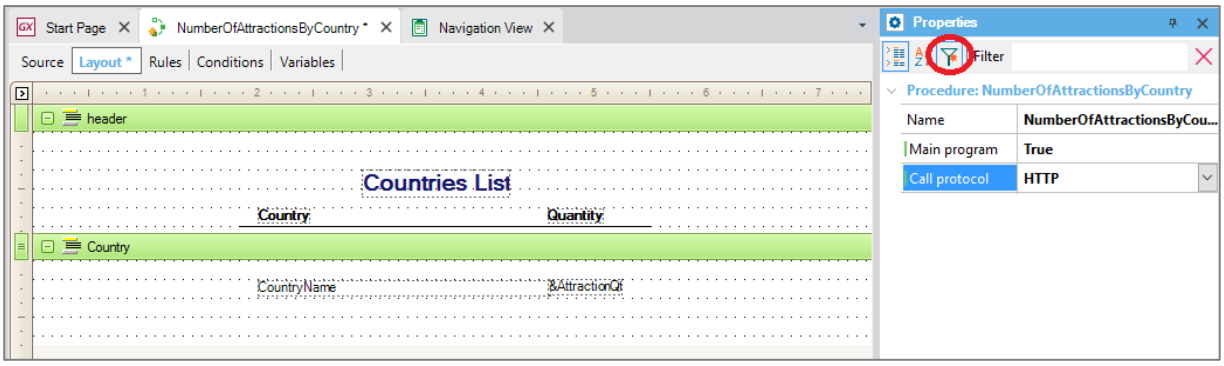


La agregamos…

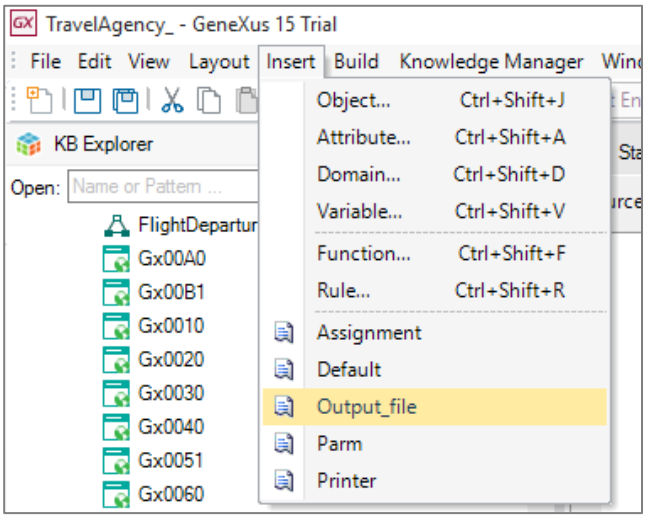


Ahora ejecutemos el reporte. Antes, recordemos configurar las propiedades para emitirlo en ambiente web y formato PDF.

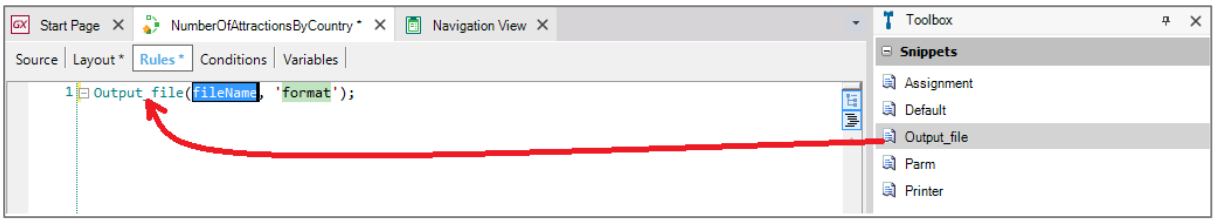
Presionando aquí podemos ver, de todas las propiedades, solo las que modificamos:



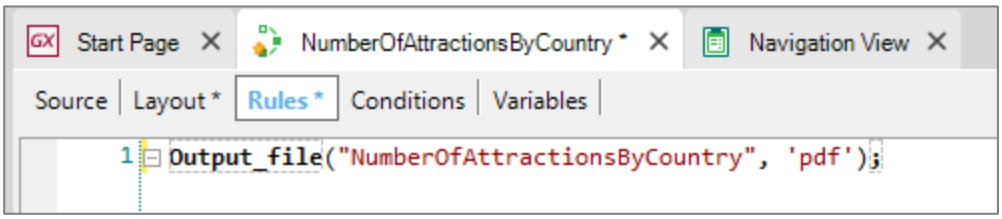
Nos falta incluir la regla output_file:
Si no la recordamos podemos buscarla desde la opción Insert del menú:



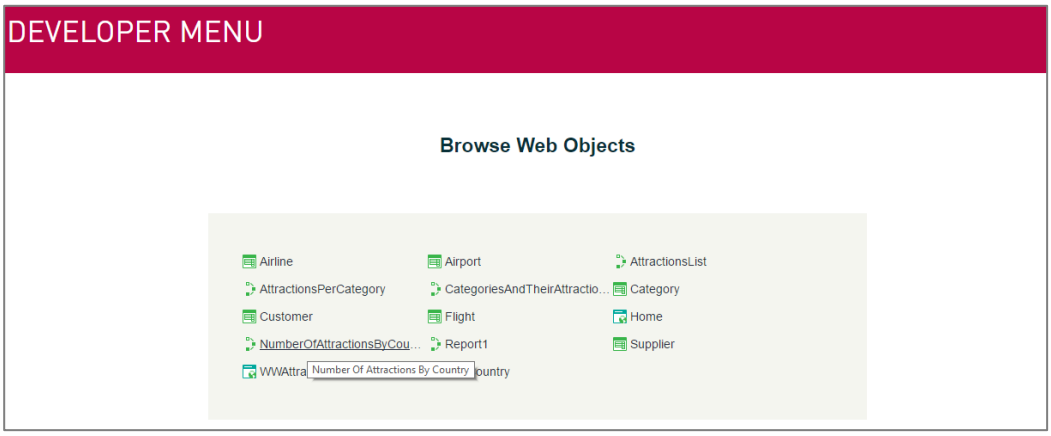
O desde la Toolbox, arrastrándola:

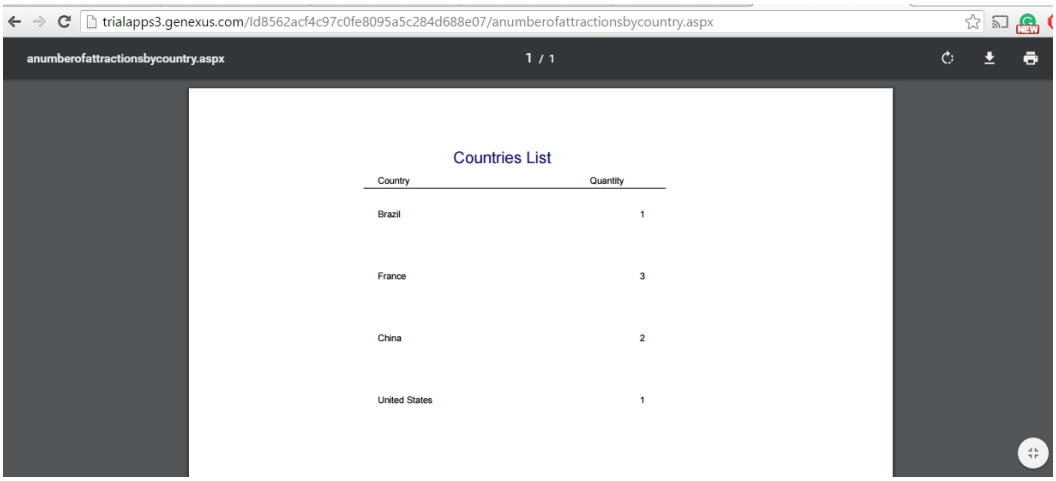


Y sustituyendo el nombre del archivo… le pondremos el mismo nombre del objeto…
Y el formato… “pdf”.



Grabamos y ejecutamos, F5:



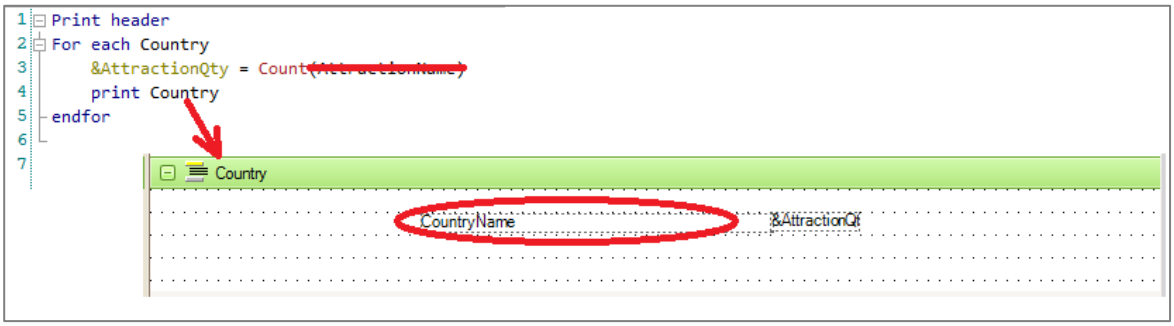


De esta manera hemos resuelto el requisito que nos solicitaron.

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.

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

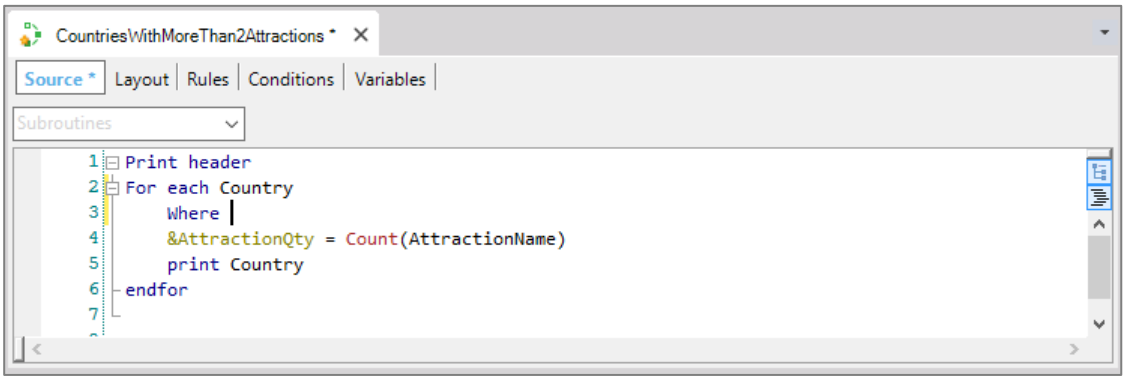
Y de igual manera, para determinar la tabla a ser navegada por el For each, se tuvieron en cuenta todos los atributos del for each **menos** los referenciados dentro de la fórmula.



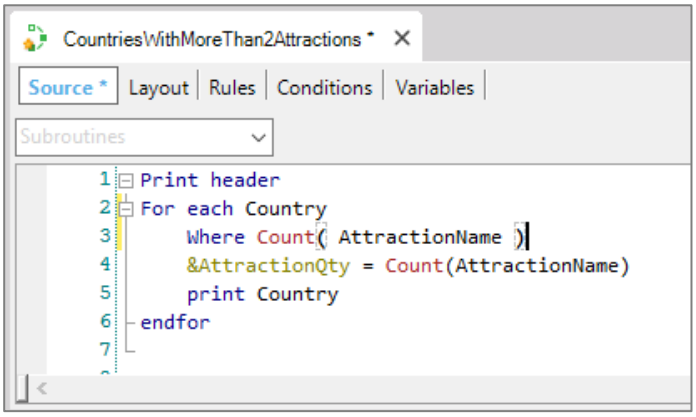
Muy bien… Supongamos que ahora nos piden que listemos todos los países que tienen **más de dos atracciones para visitar**.

Salvemos este procedimiento con otro nombre

Dado que la condición que nos piden que se cumpla para mostrar los países es que tengan más de dos atracciones turísticas, agregaremos una cláusula Where al For Each…



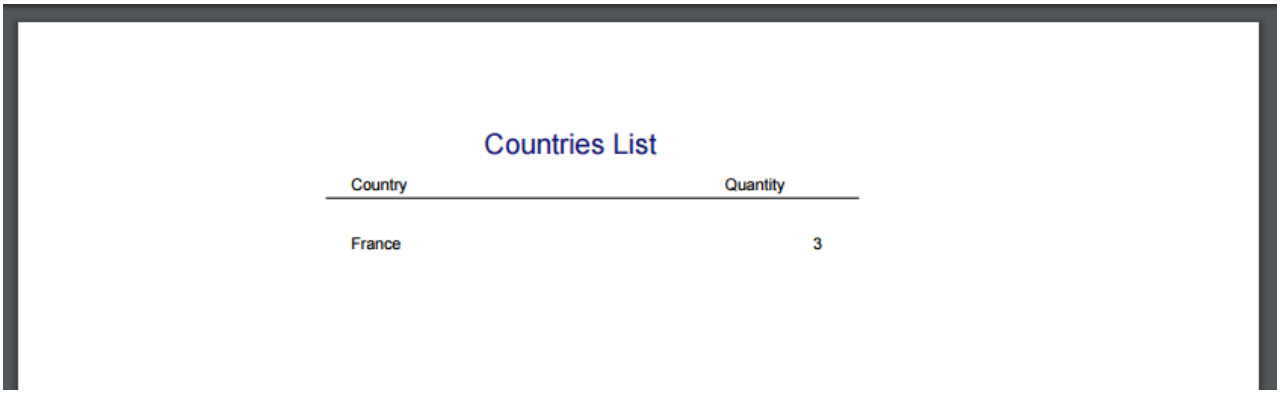
…donde queremos filtrar aquellos países para los cuales la cantidad de atracciones:



es mayor que 2:

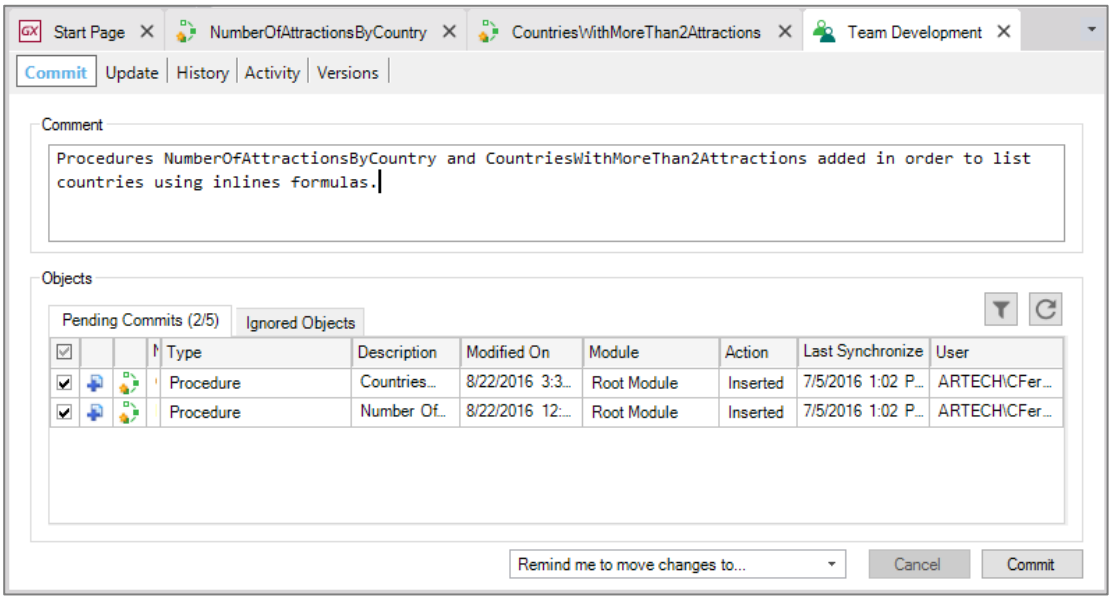
Para los que cumplan esa condición, seguiremos imprimiendo lo mismo: el nombre del país, y la cantidad de atracciones (que serán, necesariamente, mayores a dos).

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



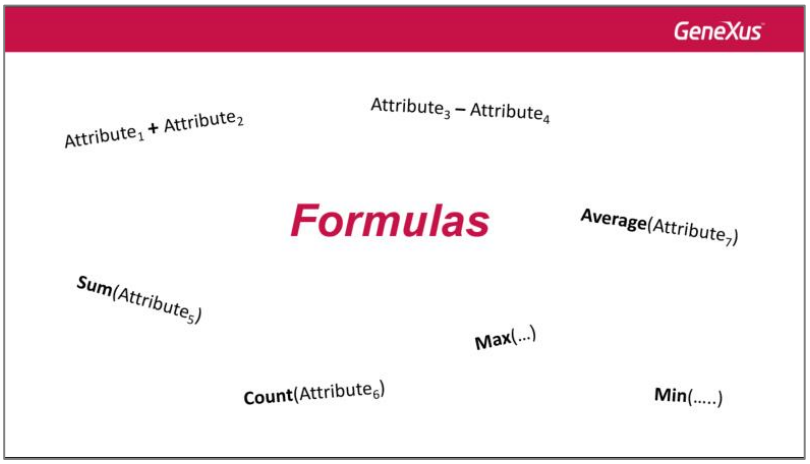
Sólo aparece Francia, con tres atracciones, como esperábamos.

Actualicemos los cambios en GeneXus Server:

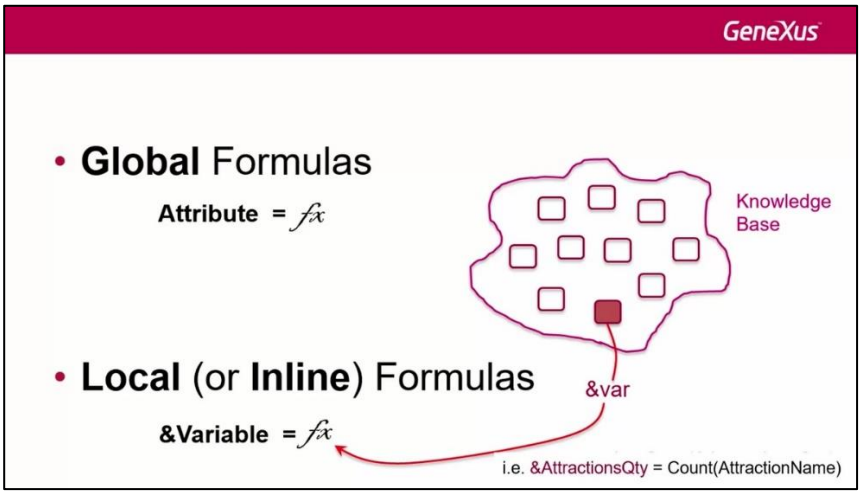


Con esto completamos dos ejemplos de uso de fórmulas “inline”, para obtener cálculos fácilmente.

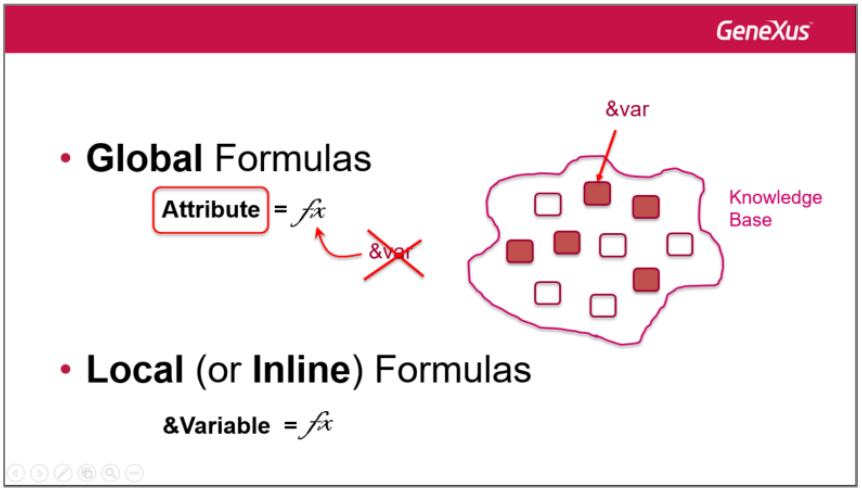
En este ejemplo hemos visto únicamente la fórmula Count, pero podríamos haber utilizado como fórmulas inline todas las que estudiamos antes, como Sum, Average, Max, entre otras.



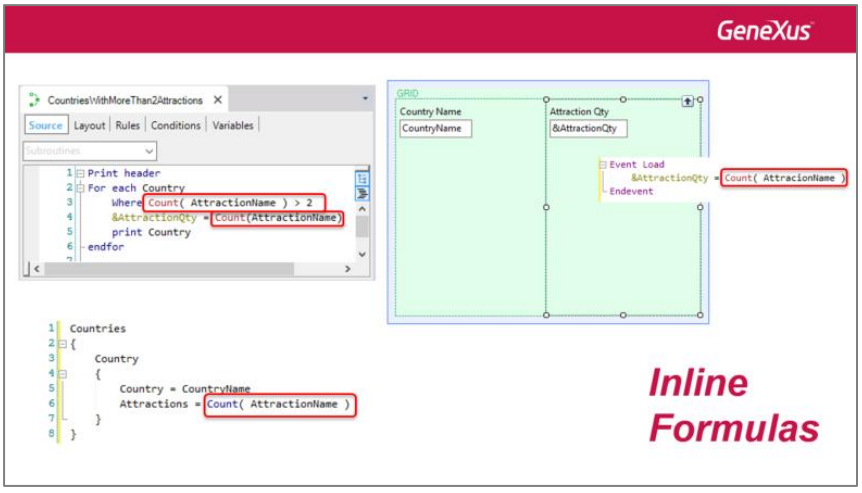
Algo importante a tener en cuenta es que como las fórmulas inline se calculan únicamente en el objeto en el que están escritas, pueden incluir dentro de su cálculo variables que se hayan definido en ese objeto. Por ejemplo…



A diferencia de lo que ocurre con las fórmulas globales, en las que no está permitido el uso de variables para el cálculo porque **son atributos** que pueden utilizarse en cualquier objeto y las variables tienen alcance únicamente local.

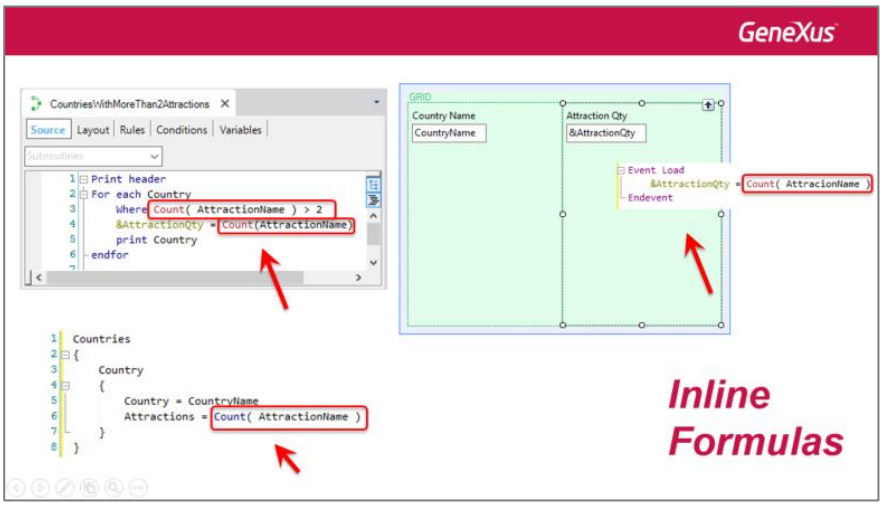


Resumiendo lo visto:



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, en el Source de un Data Provider, etc.

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



Por esa razón también le llamamos fórmula local. Se calcula en el momento en que se ejecuta el objeto y luego su valor desaparece.

Al contrario de lo que ocurría con las fórmulas globales (las definidas para atributos en transacciones), que se calculan cada vez que se consulta el valor de un atributo dentro de cualquier objeto en ejecución.