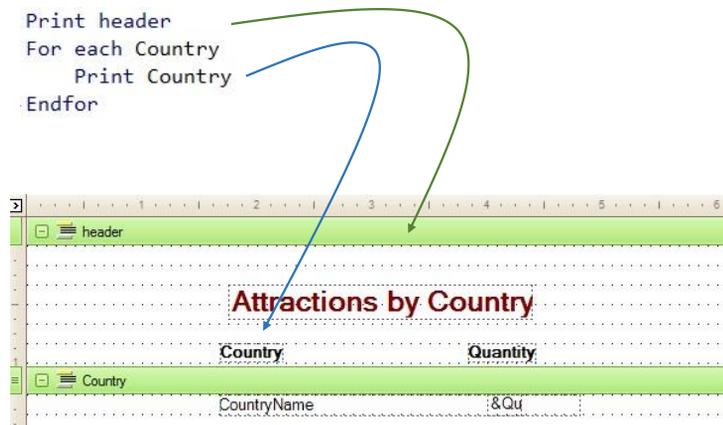


Fórmulas inline

GeneXus™

Creamos un listado para resolver el requerimiento

- Nos solicitan un listado de países con la cantidad de atracciones de cada uno
- Creamos un objeto procedimiento...



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.

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.

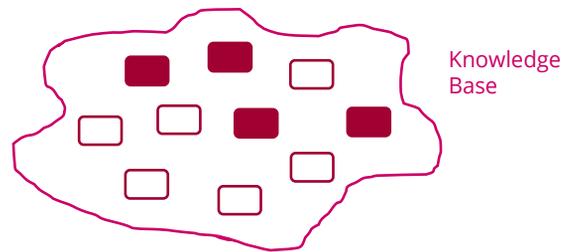
Fórmulas globales vs. fórmulas inline (locales)

- Fórmulas **Globales**

Atributo = fX



(en la estructura de la Transacción)



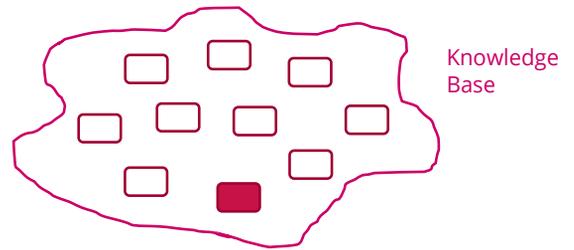
- Cualquier objeto de la KB puede acceder al cálculo
- La fórmula se evalúa cada vez que un objeto usa el atributo
- El atributo deja de estar almacenado en la base de datos

Fórmulas globales vs. fórmulas inline (locales)

- Fórmulas **Inline** (o **locales**)

$\&\text{Variable} = fX$

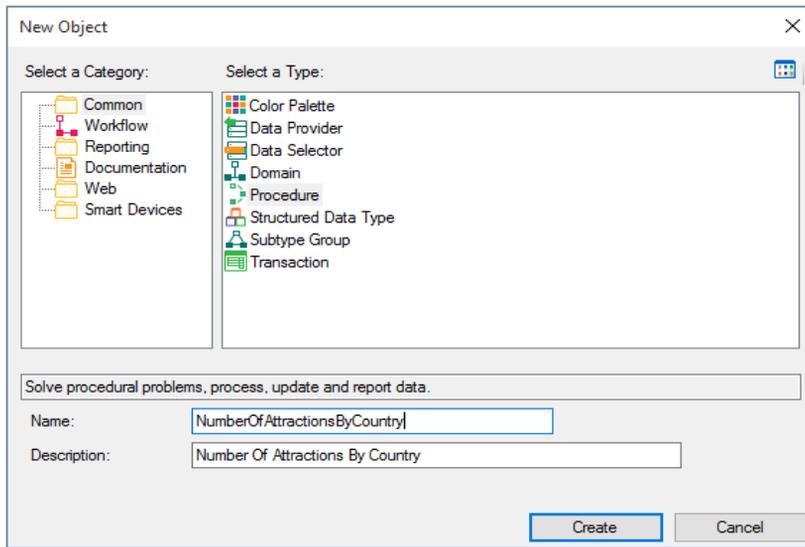
(en el código de un objeto)



- Solamente tiene acceso al cálculo el objeto que lo tiene definido
- Son el equivalente a una función, que devuelve un valor

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

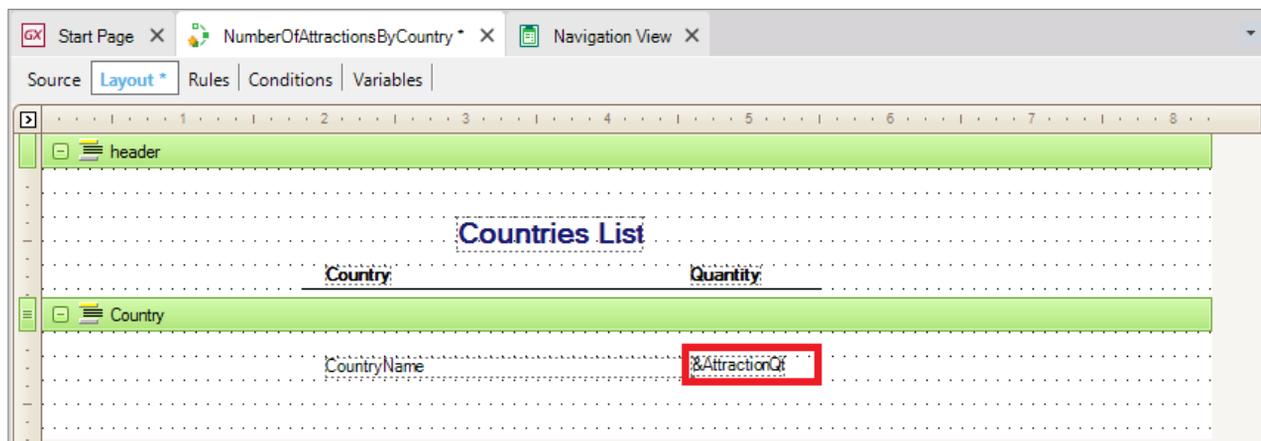
Veamos un ejemplo...



Calcularemos la cantidad mediante una fórmula "inline"... y mostraremos el resultado.

Definimos su layout

- ...con una variable &AttractionQty para mostrar el número de atracciones



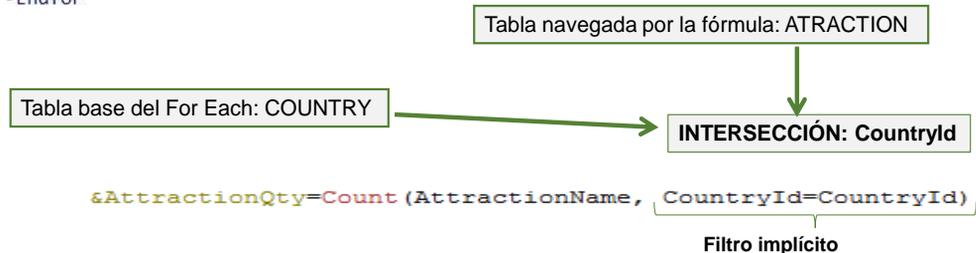
Vamos a definir una variable para asignarle el resultado que devuelva una fórmula Count y luego mostraremos la variable en el printblock. La creamos, con nombre AttractionQty, de tipo Numeric(4).

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.

Definimos una fórmula inline

| Name | Type | Is Collection | Description |
|--------------------|--------------|--------------------------|----------------|
| Variables | | | |
| Standard Variables | | | |
| AttractionQty | Numeric(4,0) | <input type="checkbox"/> | Attraction Qty |

```
Print header
For each Country
  &AttractionQty = Count(AttractionName)
  Print Country
-Endfor
```



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 ATRACCIÓN, 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.

Ejecutamos el procedimiento y vemos cómo se calcula la fórmula

Countries List

| Country | Quantity |
|---------------|----------|
| Brazil | 1 |
| France | 3 |
| China | 2 |
| United States | 1 |

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.

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. 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.

Otro uso de fórmula inline

- Requisito: listar todos los países que tienen más de 2 atracciones para visitar
- Agregaremos una cláusula Where al For Each y usamos la fórmula para la condición:

```
Print header
For each Country
  Where Count(AttractionName) > 2
  &AttractionQty = Count(AttractionName)
  Print Country
Endfor
```

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).

En ejecución...

- También pueden usarse como fórmulas inline las que vimos antes: Sum, Average, Max y otras.

Countries List

| Country | Quantity |
|---------|----------|
| France | 3 |

Algo que vale la pena recordar...



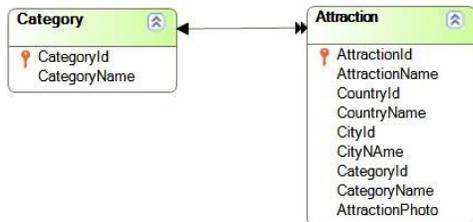
Dado que este procedimiento fue grabado a partir del otro, ya tiene configurado todo lo necesario para ser impreso como PDF, así que lo ejecutamos y vemos el resultado.

Sólo aparece Francia, con tres atracciones, como esperábamos.

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.

Fórmulas inline + cláusula Unique

Permite también la unión de información con fórmulas inline.



| Categories in Attractions | |
|---------------------------|-----|
| CategoryName | &Qu |

Devuelve la lista (sin elementos repetidos) de todas las categorías, cada una con la correspondiente cantidad de atracciones registradas.

```

Print Title
For each Attraction
  Unique CategoryId
  &Quantity = Count(AttractionId)
  Print Categories
Endfor
  
```

Tabla base del For each:
ATTRACTION

Tabla base de la fórmula:
ATTRACTION

Hemos visto anteriormente el concepto y un ejemplo de aplicación de la cláusula Unique dentro de un For each.

Habíamos definido un listado mostrando las categorías que tienen atracciones turísticas registradas, pero sin que se repitan los nombres de esas categorías. Vamos a agregar ahora un requerimiento más: Junto a cada nombre de categoría queremos ver su cantidad de atracciones.

Para calcular la cantidad de atracciones, declaramos ahora la siguiente fórmula inline.. &Quantity=Count(AttractionId)

Observemos que la tabla base del For each es ATTRACTION, al igual a la tabla base de la fórmula inline.

Por lo tanto, la fórmula Count agregará del contexto una condición implícita en su evaluación: Contará todas las atracciones para el atributo declarado en la cláusula unique.

En la navegación se muestra que la fórmula contará todas las instancias de atracciones para el atributo CategoryId declarado (Given) en la cláusula unique.

Fórmulas globales vs. fórmulas inline (locales)

- Las fórmulas inline pueden usar variables del objeto donde fueron definidas.

Fórmulas Inline (o locales)

`&Variable = fX`



Knowledge
Base

p.ej. `&AttractionsQty = Count(AttractionName)`

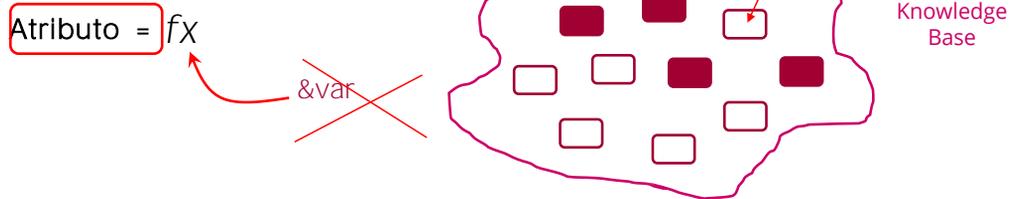
Algo importante a tener en cuenta es que como las fórmulas inline se calculan únicamente en el objeto en el que están escritas, pueden incluir dentro de su cálculo variables que se hayan definido en ese objeto.

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.

Fórmulas globales vs. fórmulas inline (locales)

- En las fórmulas globales 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.

Fórmulas Globales



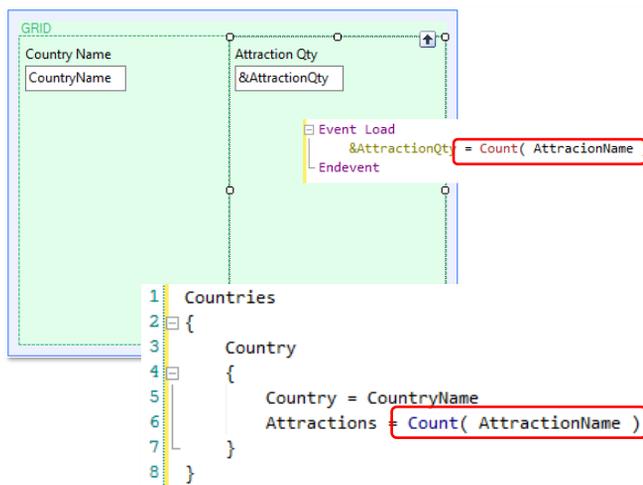
Resumiendo...

- 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.

```

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

```



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.

Resumiendo...

- 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.

*GeneXus*TM

training.genexus.com
wiki.genexus.com