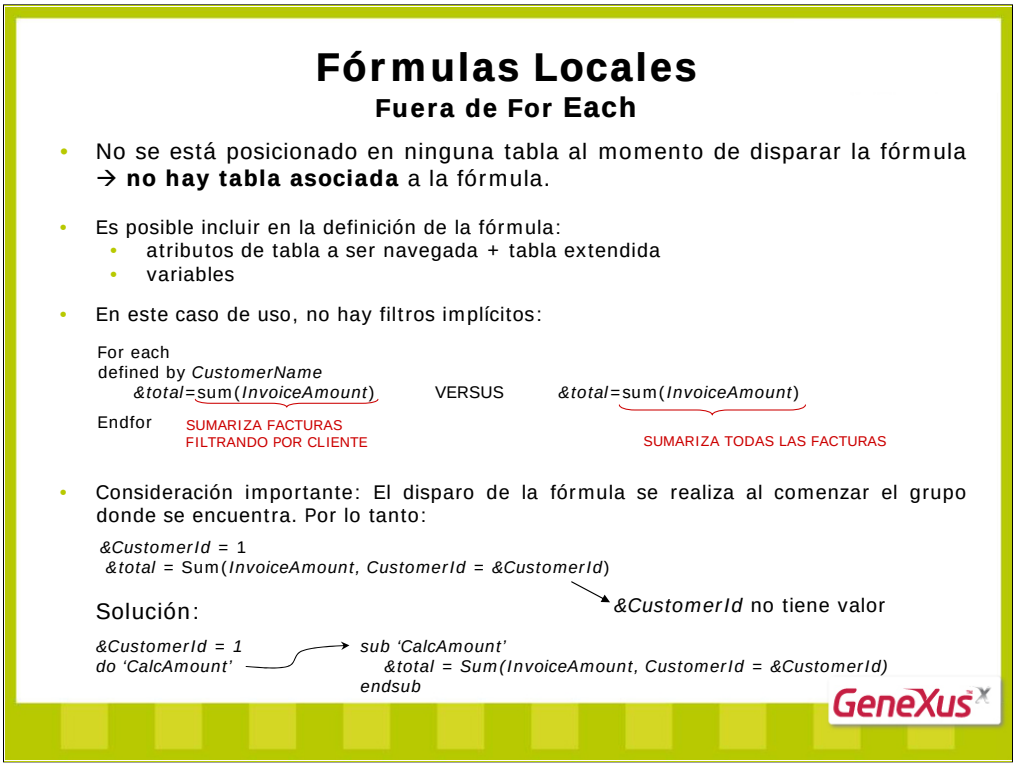
Fórmulas locales

Fórmulas Locales
Generalidades

- Además de definir fórmulas asociadas a atributos (globales, a nivel de la KB) **también es posible definir fórmulas en el propio código.**
- Llamamos a este caso de uso de las fórmulas: **fórmulas inline o fórmulas locales.**
- Es posible definir fórmulas inline = locales → en source de procedimientos, subrutinas, eventos, etc.
- Comportamiento depende de si se definen:
  - Dentro de For Each
  - Fuera de For Each

Otro ejemplo de codificación posible es el que sigue, en el cual la fórmula se define en el propio where del For Each, para filtrar por dicho cálculo:

For each order *CountryId*
where **Count(*CustomerName*)>50**
…
Endfor



Consideración importante

Una fórmula se dispara cuando comienza el grupo que la contiene.

Esto significa que si se tiene definida una fórmula con sus parámetros en determinada línea de código, los valores de los parámetros van a ser aquellos que fueron leídos cuando el grupo fue ejecutado en la base de datos. Entonces, si el grupo fue ejecutado en la base de datos, y luego de eso ud. asignó valores diferentes a los parámetros que involucrará en la definición de la fórmula, los valores de los parámetros que se tendrán en cuenta no serán los que usted asignó; por el contrario serán los leídos cuando se ejecutó el grupo que contiene a la fórmula en la base de datos.

¿Y cuándo se ejecutan los grupos?

- Cuando comienza un programa.
- Cuando comienza un For Each.
- Luego de un Endfor.
- Cuando comienza una subrutina o un evento.

En particular como en este caso estamos explicando **fórmulas locales definidas fuera de comandos For Each**, los ítems que aplican son 1, 3 y 4.

Lo mismo ocurre con las variables. Los valores de las variables que son tenidos en cuenta al momento de disparar las fórmulas, son los valores que las variables tienen asignados cuando el grupo que contiene a la fórmula se ejecuta. De modo que si escribimos el siguiente código:

&CustomerId = 1
&total = Sum(*InvoiceTotal, CustomerId = &CustomerId*)

La variable *&CustomerId* no tendrá valor cuando la fórmula se dispare.

Esto puede resolverse como sigue:

&CustomerId = 1	sub ‘AmountCalc‘
do ‘AmountCalc‘	*&total* = Sum(*InvoiceAmount*, *CustomerId* = *&CustomerId*) endsub

ya que así logramos un nuevo comienzo de grupo (cuando la subrutina comienza), y en ese momento la variable ya está cargada con el valor deseado. De modo que con esta solución, cuando la fórmula se ejecute la variable *&CustomerId* valdrá 1.