

GeneXus X Evolution 3

Fórmulas

Clasificación

- Fórmulas Aggregate no se dividen más en:

- Verticales
- Aggregate/Select

Clasificación actual de fórmulas:

- **Horizontales:** Una o varias expresiones aritméticas
- **Aggregate:** Sum, Count, Average, Max, Min, Find
- **Compuestas:** Conjunto de expresiones horizontales y/o aggregate

Alcance

- **Antes:** Fórmulas Aggregate permitían involucrar atributos de:
 - Tabla de partida y su extendida
(tabla en la cual se está posicionado al momento de disparar la fórmula + extendida)
 - Tabla de llegada
(tabla en la cual se evalúa la fórmula... no incluyendo su extendida)
- **Ahora:** Fórmulas Aggregate permiten involucrar atributos de:
 - Tabla de partida y su extendida
 - Tabla de llegada y su extendida

Fórmulas inline

Además de definir fórmulas asociadas a atributos (globales, a nivel de la KB) ¡también es posible definir fórmulas en el propio código!

- Es posible definir fórmulas inline = locales → en source de procedimientos, subrutinas, eventos, etc.
- Comportamiento depende de si se definen:
 - 1) Dentro de For Each
 - 2) Fuera de For Each

Fórmulas inline dentro de For each

- Tabla de partida de la fórmula = Tabla base del For Each
- Es posible incluir en la definición de la fórmula:
 - atributos de tabla de partida + extendida
 - atributos de tabla de llegada + extendida
 - variables

```
For each Country
  Where count(CustomerId) > 10
  -----
Endfor
```

TABLA BASE FOR EACH = COUNTRY
TABLA PARTIDA FÓRMULA = COUNTRY
TABLA LLEGADA FÓRMULA = CUSTOMER
ATRIBUTO EN COMÚN: *CountryId*

```
Country
{
  CountryId*
  CountryName
}
```

```
Customer
{
  CustomerId*
  CustomerName
  CountryId
  CountryName
}
```

Fórmulas inline fuera de For each

- No se está posicionado en ninguna tabla al momento de disparar la fórmula → no hay tabla de partida para la fórmula
- Es posible incluir en la definición de la fórmula:
 - atributos de tabla de llegada + extendida
 - variables
- En este caso de uso, no hay filtros implícitos:

```
For each Country
  &Quantity = count(CustomerId)
  -----
```

```
Endfor
```

**CUENTA CLIENTES
FILTRANDO POR PAIS**

vs **&Quantity = count(CustomerId)**

CUENTA TODOS LOS CLIENTES

```
Country
{
  CountryId*
  CountryName
}
```

```
Customer
{
  CustomerId*
  CustomerName
  CountryId
  CountryName
}
```

Consideración importante: La fórmula es disparada cuando el grupo que la contiene comienza.

Esto significa que si se tiene la fórmula definida dentro de una cierta línea del código con sus parámetros, los valores de éstos serán aquellos que fueron leídos al momento en que el grupo fue ejecutado en la base de datos. Por lo tanto, si el grupo se ejecutó en la base de datos y luego se le asignaron valores diferentes a los parámetros que estarán involucrados en la definición de la fórmula, los valores de los parámetros que se tomarán en cuenta no serán aquellos que se asignaron; por el contrario, serán los leídos cuando el grupo que contenía la fórmula fue ejecutado en la base de datos.

Por lo tanto, ¿cuándo son ejecutados los grupos?

- cuando un programa comienza
- cuando un For Each comienza
- luego de un Endfor
- cuando una subrutina o evento comienzan

Como en este caso estamos explicando **fórmulas inline que son definidas fuera de comandos For Each**, los ítems que aplican con los 1, 3 y 4.

Lo mismo sucede con las variables. Los valores de las variables que son tomados en cuenta cuando las fórmulas son disparadas, son los valores que las variables han tenido asignados al momento en que el grupo que contiene la fórmula es ejecutado. Por tanto, se puede escribir el siguiente código:

```
&CustomerId = 1
```

`&total = Sum(InvoiceTotal, CustomerId = &CustomerId)`

La variable `&CustomerId` no tendrá un valor al momento en que la fórmula es disparada.

Esto puede ser resuelto como sigue:

```
&CustomerId = 1      sub 'calctotal'
do 'calctotal'        &total = Sum(InvoiceTotal, CustomerId = &CustomerId)
                      endsub
```

En esta forma, como hay un nuevo comienzo de un grupo (cuando comienza la subrutina) y la fórmula está incluida dentro de la subrutina, cuando la fórmula sea ejecutada la variable ya tendrá el valor deseado.

Ejemplos

Pueden utilizarse en órdenes y en filtros

```
For Each Customer order CustomerAge
-----
Endfor
```

atributo fórmula en order

```
For each Country
  where Count(CustomerName) > 50
-----
Endfor
```

fórmula definida en el propio where

```
Country
{
  CountryId*
  CountryName
}
```

```
Customer
{
  CustomerId*
  CustomerName
  CustomerAge
  CountryId
  CountryName
}
```

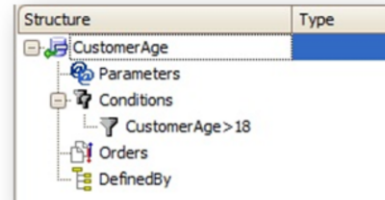
Uso de Data Selectors en Fórmulas

Es posible incluir un Data Selector en la definición de una fórmula Aggregate, **mediante la cláusula USING...**

```
Country
{
  CountryId*
  CountryName
  CountryCustomerAgeAverage
}
```

```
Customer
{
  CustomerId*
  CustomerName
  CustomerAge
  CountryId
  CountryName
}
```

Average(CustomerAge, USING CustomerAge())



Sobre el objeto Data Selector volveremos más adelante.

GeneXus™