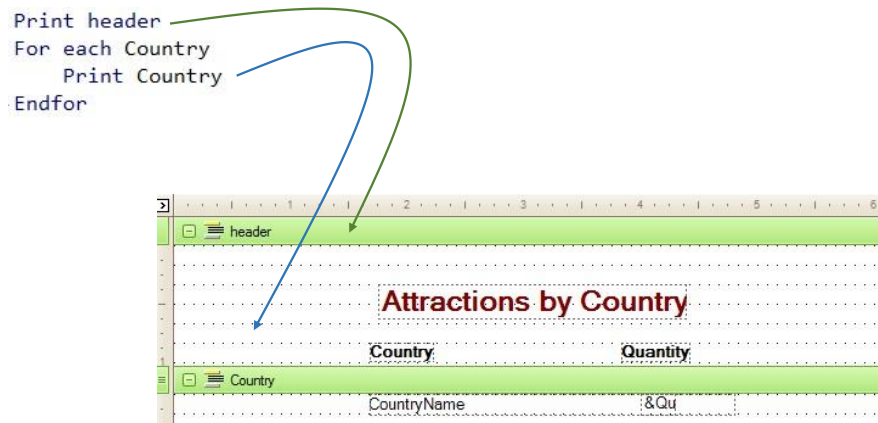


Fórmulas inline

GeneXus™

Criamos uma listagem para resolver o requisito

- Foi solicitada uma listagem de países com a quantidade de atrações de cada um
- Criamos um objeto procedure...



Em uma aula anterior já havíamos diferenciado entre fórmulas globais e fórmulas locais ou inline. As fórmulas globais eram definidas nos atributos na estrutura de uma transação, ou seja, especificava-se que o atributo era sempre calculado, isso fazia que em qualquer objeto onde fosse necessário recuperar o valor do atributo a fórmula era avaliada para dar o seu resultado. Por isso, os atributos fórmulas eram atributos virtuais, que não eram armazenados nas tabelas.

Em contra-partida, as fórmulas também podem ser utilizadas de forma local, isto é, especificadas para serem avaliadas somente no lugar do código do objeto onde se encontram. Por exemplo, no Source de um procedimento. Poderíamos, por exemplo, atribuir a uma variável o resultado de uma fórmula.

Vejamos alguns exemplos de uso.

Suponhamos que a agência de viagens solicita um relatório que mostre todos os nomes de países e para cada país a quantidade de atrações turísticas disponibilizadas. Para resolver essa solicitação, vamos criar um objeto procedimento.

Posicionemos no Source do procedimento e a primeira coisa que faremos é imprimir um cabeçalho com um título, assim escrevemos a instrução: Print header. Header será um printblock que ainda não temos. Então, vamos defini-lo.

Vamos na seção Layout, e no printblock que temos por padrão alteramos o nome para "header".

Podemos agora inserir um textblock e dar-lhe o formato desejado, ou irmos até o relatório de atrações que tínhamos definido anteriormente, e fazer copy&paste do textblock para manter seu formato, e mudar somente a propriedade Text. Podemos ver que foi adicionado no printblock dois textos para as colunas do relatório e uma linha.

Voltemos ao Source para continuarmos implementando o relatório. Na continuação da instrução que imprime o cabeçalho, devemos escrever um comando For each para acessar a tabela que armazena os países e mostrá-los. Dentro do For each escrevemos: Print Country... com o objetivo de mostrar no printblock o atributo CountryName. Temos que criar esse printblock, e dentro adicionamos o atributo CountryName alinhando-o com a coluna do título Country. Voltemos ao Source. Esse For each definido lerá toda a tabela de países e mostrará o nome de cada país consultado.

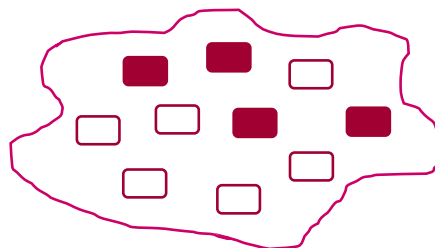
Fórmulas Globais vs. Fórmulas Inline (local)

• Fórmulas **Globais**

$$\text{Atributo} = fX$$



(na estrutura da Transação)



Knowledge
Base

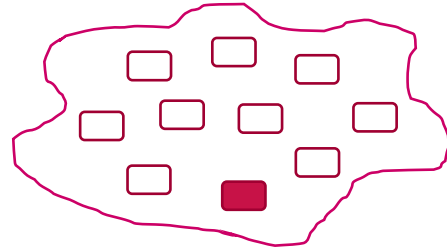
- Qualquer objeto da KB pode acessar o cálculo.
- A fórmula é disparada cada vez que o objeto usa o atributo.
- O atributo deixa de ser armazenado na base de dados.

Fórmulas Globais vs. Fórmulas Inline (local)

- Fórmulas **Inline** (ou **locais**)

$$\underbrace{\&Variable = fX}$$

(no código de um objeto)

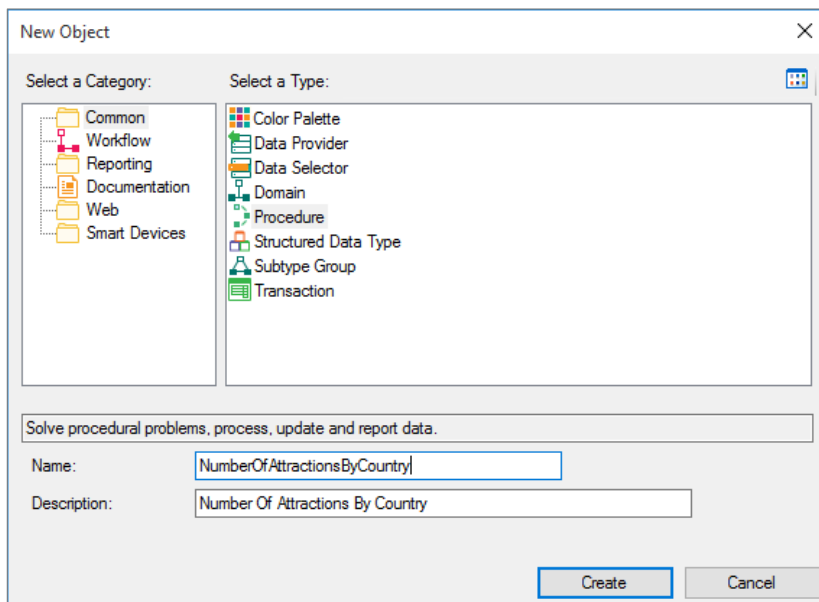


- O cálculo está disponível somente para o objeto onde foi definido..
- São o equivalente a uma função que retorna um valor.

Agora somente nos falta resolver o requisito de que para cada país seja impresso a quantidade de atrações que possui. Como podemos implementar isso?

Vejamos um exemplo...

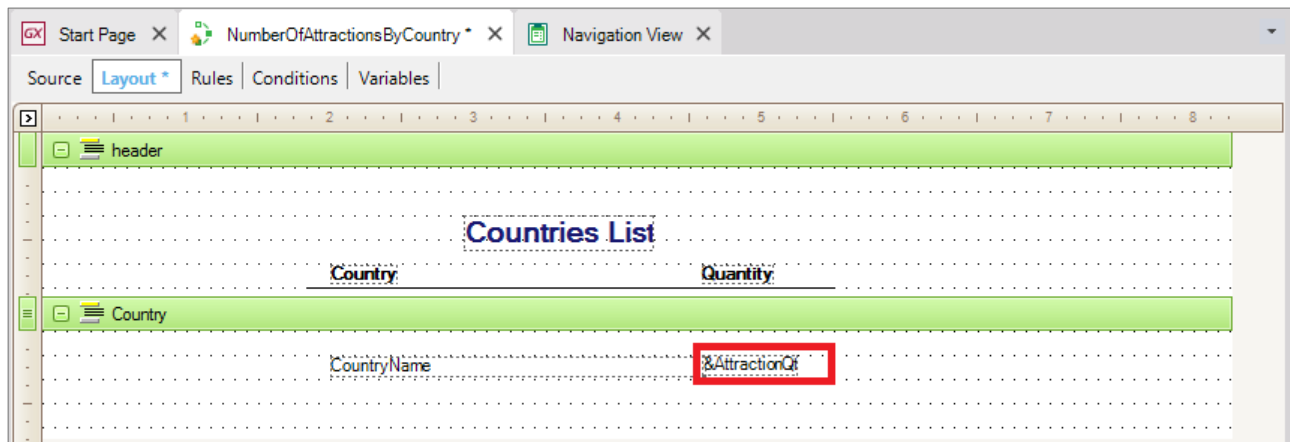
- Foi solicitada uma listagem de países com a quantidade de atrações de cada um deles.
- Nós criamos um objeto procedure ...



Calcularemos a quantidade com uma fórmula "inline" e mostraremos o resultado.

Nós definimos seu layout

- ...com uma variável &AttractionQty para mostrar o número de atrações.



Vamos definir uma variável para atribuir o resultado devolvido por uma fórmula Count e depois mostraremos a variável no printblock. Criamos a variável com nome AttractionQty, do tipo Numeric(4).

Lembremos que as variáveis são espaços de memória locais no objeto onde são definidas. Isso quer dizer que só existem durante cada execução desse objeto. Quando a execução termina, desaparecem.

Nós definimos uma 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
```

Tabela navegada pela fórmula: ATTRACTION

Tabela Base para o comando For Each : COUNTRY

INTERSECÇÃO: CountryId

```
&AttractionQty=Count (AttractionName, {CountryId=CountryId})
```

Filtro Implícito

Voltemos ao Source e dentro do For each, antes de imprimir, colocamos a variável: &AttractionQty... e atribuímos o resultado de uma fórmula Count... que vai contar as atrações.

Dentro do parênteses devemos incluir um atributo que faça com que GeneXus saiba qual tabela queremos contar. Como é a tabela ATTRACTION, escolheremos algum atributo que esteja nesta tabela. Por exemplo, AttractionName.

Analizemos esse código para ver como funciona. As fórmulas determinam a tabela a ser navegada pelos atributos referenciados dentro do parênteses. Nesse caso temos incluído o atributo AttractionName, porque queremos contar atrações e GeneXus entende exatamente assim. Mas, as atrações que queremos contar não são todas da tabela, senão as que correspondem ao país que estamos posicionados em cada execução do For each. Se a fórmula não estivesse dentro de um For each, nem em um contexto no qual se possa estar posicionada em uma tabela, seriam contadas todas as atrações. Como a fórmula está definida dentro de um comando For each, ou seja, encontra-se em um contexto no qual está lendo uma tabela, nesse caso a tabela de países, será influenciada por esse contexto. Assim, não será contadas TODAS as atrações da tabela, apenas as relacionadas com o país que está sendo processado em cada iteração do for each. Em outras palavras: GeneXus determina a tabela que deverá ser lida na fórmula com base nos seus atributos, e analisa se há relação entre essa tabela e a tabela base do For each (incluindo sua estendida).

Podemos observar que nesse caso há um atributo em comum entre as duas tabelas, CountryId, assim que para cada país que o For each encontra, será contada suas atrações.

Executamos o procedimento e vemos como a formula foi calculada

Countries List

Country	Quantity
Brazil	1
France	3
China	2
United States	1

Assim, se cadastramos nas transações que já tínhamos definidos: o Museu Matisse da França e a cidade proibida da China:

O for each é executado, e para o primeiro país, Brazil, são contadas as atrações com esse mesmo CountryId...

Na sequência, o for each passa ao seguinte registro, 2-France, e conta suas atrações, as de CountryId=2... encontra três.

Logo itera o seguinte registro, 3 – China, contando 2 atrações.

E por último United States, com uma só atração.

Em resumo, funciona como se estivéssemos definido essa condição de filtro explícita na fórmula:

Onde um CountryId é o da tabela navegada pela fórmula e o outro é o da tabela do for each. Não é necessário escrever a condição. GeneXus detecta automaticamente. Outra coisa detectada automaticamente, é que GeneXus concordará em ler a tabela a ser navegada pela fórmula ordenando-a por esse atributo de filtro. Do contrário, como vemos nesse exemplo, para cada país teria que ler toda a tabela de atrações para contar as que correspondem, a cada vez.

Podemos ver todas essas inferências que GeneXus realiza através do relatório de navegação.

Podemos ver que o For each lerá a tabela Country, e para cada país realizará o Count, pelo que deverá navegar a tabela Attraction por CountryId. Somente resta-nos adicionar a variável &AttractionQty calculada como primeira instrução no corpo do For each no printblock Country, para que imediatamente depois de ter o resultado da contagem na variável, ela seja

exibida ao lado do nome do país.

Adicionamos... Agora executemos o relatório. Antes, lembremos de configurar as propriedades para emití-lo em ambiente web e formato PDF. Pressionando o ícone destacado na figura abaixo, podemos ver as propriedades que alteramos: Falta incluir a regra Output_file: Se não lembrarmos, podemos buscar mediante a opção Insert do menu: Ou via Toolbox, arrastando-a e substituindo o nome do arquivo... colocaremos o mesmo nome do objeto... e o formato... "pdf".

Salvamos e executamos com F5:

Dessa forma, temos resolvido o requisito que foi solicitado. Como temos estudado, para determinar a tabela a ser navegada pela fórmula não se leva em conta os atributos referenciados no For each, senão somente os atributos incluídos na definição da fórmula. E de igual maneira, para determinar a tabela a ser navegada pelo For each, foi levado em conta todos os atributos do for each menos os referenciados dentro da fórmula.

Outro uso da fórmula inline

- Requisito : listar todos os países que tenham mais de 2 atrações para visitar.
- Adicionaremos uma cláusula Where no comando For Each e usaremos a fórmula para a condição:

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

Muito bem... Suponhamos que agora seja solicitado que listemos todos os países que possuem mais de duas atrações para visitar. Vamos salvar esse procedimento com outro nome. Dado que a condição que foi solicitada pede que mostremos os países que tenham mais de duas atrações turísticas, adicionaremos uma cláusula Where no For Each... onde queremos filtrar os países para os quais a quantidade de atrações: é maior que 2.

Para implementar essa condição, continuaremos imprimindo o mesmo: o nome do país, e a quantidade de atrações (que serão, necessariamente, maiores que dois).

Em tempo de execução (runtime) ...

Countries List

Country	Quantity
France	3

Lembre que:

- As fórmulas que vimos antes (Sum, Average, Max e outras) também podem ser usadas como Fórmulas Inline.

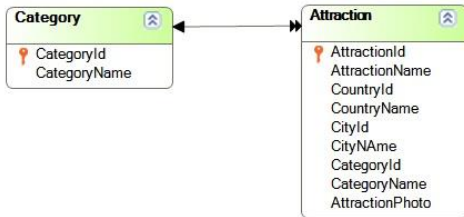


Dado que esse procedimento foi criado a partir de outro, ele já tem configurado tudo o que é necessário para ser impresso como PDF, assim executamos-os e visualizamos o resultado. Só aparece France com três atrações, como esperávamos.

Com isso completamos dois exemplos de uso de fórmula “inline” para obter cálculos facilmente. Nesse exemplo estudamos unicamente a fórmula Count, mas poderíamos ter utilizado como fórmulas inline todas as que estudamos antes, como Sum, Average, Max, entre outras.

Fórmulas inline + cláusula Unique

Permite também a união de informação com fórmulas inline.



Devolve a lista (sem elementos repetidos) de todas as categorias, cada uma com a correspondente quantidade de atrações registradas.

```

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

Tabela base do
For each: **ATTRACTION**

Tabela base da fórmula:
ATTRACTION

Vimos anteriormente o conceito e um exemplo de aplicação da cláusula Unique dentro de um For each.

Tínhamos definido uma lista mostrando as categorias que têm atrações turísticas registradas, mas sem que se repitam os nomes dessas categorias.

Agora vamos adicionar mais um requisito: Ao lado do nome de cada categoria queremos ver sua quantidade de atrações.

Para calcular a quantidade de atrações, declaramos agora a seguinte fórmula inline.. &Quantity=Count(AttractionId)

Observemos que a tabela base do For each é ATTRACTION, assim como a tabela base da fórmula inline.

Portanto, a fórmula Count adicionará a partir do contexto uma condição implícita em sua avaliação: Contará todas as atrações para o atributo declarado na cláusula unique.

Na navegação é exibido que a fórmula contará todas as instâncias de atrações para o atributo CategoryId declarado (Given) na cláusula unique.

Fórmulas Globais vs. Fórmulas Inline (locais)

- Fórmulas Inline podem usar variáveis do objeto que foram definidas.

Fórmulas **Inline**(ou **locais**)

&Variable = fX



Knowledge
Base

e.x. `&AttractionsQty = Count(AttractionName)`

Algo importante a ter em conta é que como as fórmulas inline são calculadas unicamente no objeto no qual estão escritas, podem incluir dentro do seu cálculo variáveis que são definidas nesse objeto. Por exemplo...

A diferença do que ocorre com as fórmulas globais, nas quais não se está permitido o uso de variáveis para o cálculo porque são permitidos apenas atributos, que podem ser utilizados em qualquer objeto, as fórmulas inline aceitam variáveis, lembrando que possuem alcance apenas local.

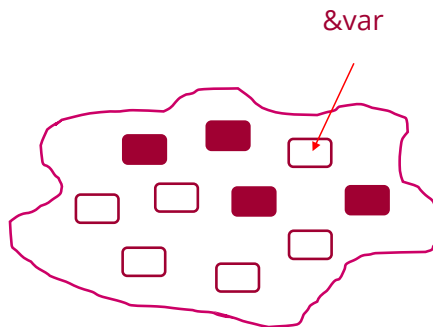
Fórmulas Globais vs. Fórmulas Inline (locais)

- Fórmulas Globais não permitem o uso de variáveis para o cálculo pois **são atributos** que podem ser utilizados por qualquer objeto e as variáveis tem escopo local apenas.

Fórmulas **Globais**

Atributo = fX

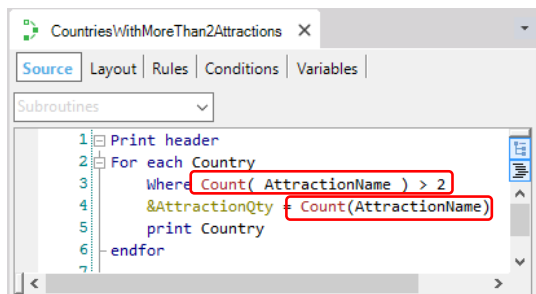
~~&var~~



Knowledge
Base

Resumindo...

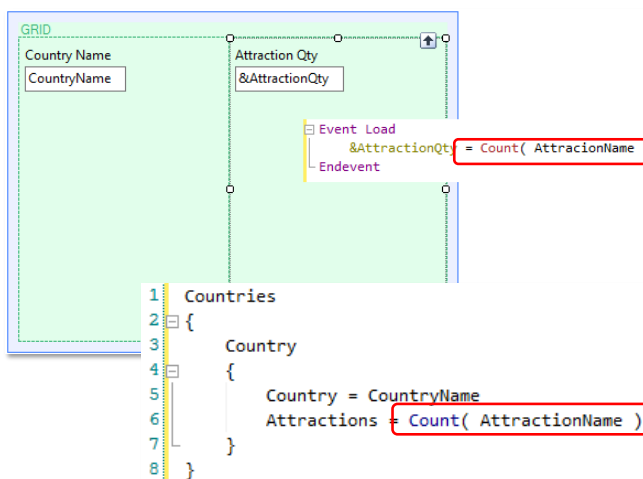
- Uma fórmula “inline” é uma fórmula que declaramos como uma instrução específica dentro um certo pedaço de código, como no Source de uma procedure, evento de um web panel, Source de um Data Provider, etc.



```

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

```



Resumindo o que foi estudado:

Uma fórmula “inline” é uma fórmula que declaramos como instrução pontual dentro de determinado código, como no Source de um procedimento, em um evento de uma Web Panel, no Source de um Data Provider, etc. A fórmula somente é conhecida no objeto onde foi definida.

Por essa razão, também são chamadas de fórmula local. São calculadas no momento em que se executa o objeto e o seu valor logo desaparece. Ao contrário do que ocorria com as fórmulas globais (as definidas para atributos em transações), que são calculadas a cada momento que se consulta o valor de um atributo dentro de qualquer objeto em execução.

Resumindo...

- A fórmula é apenas conhecida pelo objeto que a definiu.
- Por este motivo também a chamamos de fórmula local. É calculada no momento que o objeto é executado e logo após seu valor desaparece.
- São diferente das fórmulas globais (que são definidas para atributos em transações), que são calculadas toda vez que o valor de um atributo é requisitado por qualquer objeto em execução.

*GeneXus*TM

training.genexus.com
wiki.genexus.com