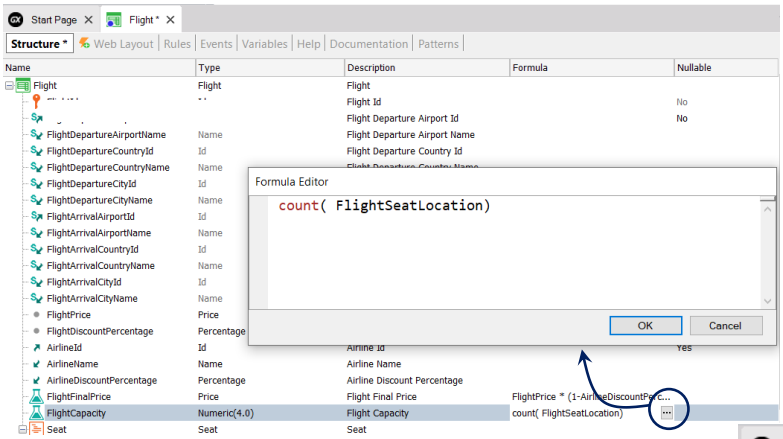
Fórmula vs. regra de atribuição

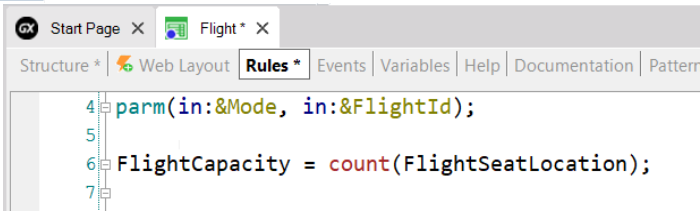
GeneXus

A seguir, vamos nos aprofundar em alguns dos conceitos das fórmulas. Comecemos vendo a diferença entre definir um atributo através de um cálculo em uma fórmula ou em uma regra de atribuição.

Considerações ao definir um atributo com base em um cálculo: regra ou fórmula?



Vs.

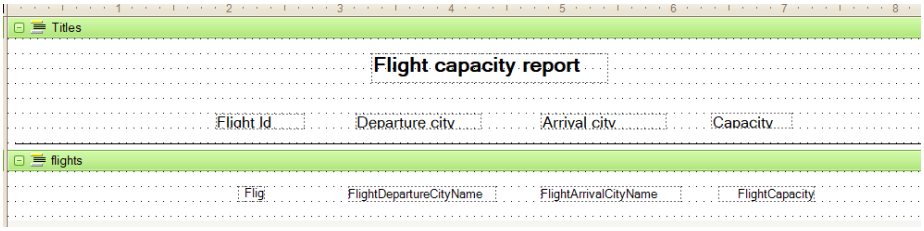


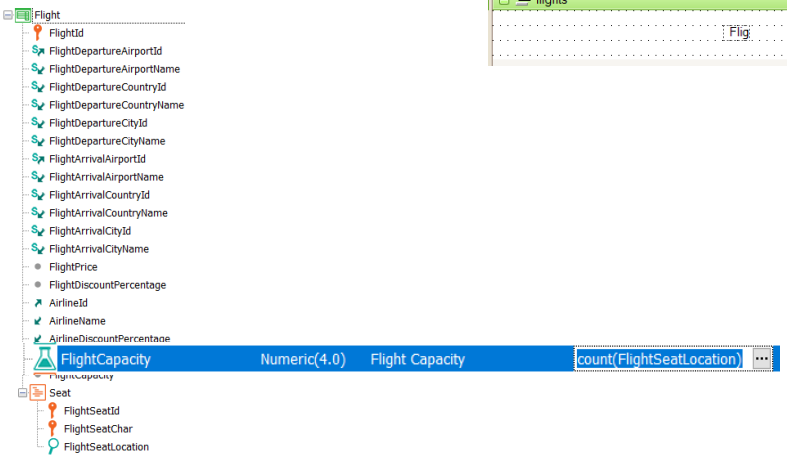
Quando o valor de um atributo pode ser obtido por meio de um cálculo, geralmente o definimos como fórmula na estrutura da transação.
No entanto, notemos que também poderíamos atribuir o cálculo ao atributo usando uma regra.

Que considerações devemos levar em conta para decidir se atribuímos o cálculo por meio de uma regra ou se o fazemos definindo o atributo como fórmula? Já vimos anteriormente que se o atributo é fórmula, GeneXus não cria em sua tabela associada (isto é, a tabela a que pertenceria o atributo fórmula se estivesse armazenado), um campo para armazenar o valor, pois entende que seu valor pode ser obtido a partir do cálculo. Por isso, dizemos que consideramos o atributo como "virtual", pois continua presente na estrutura da transação, mas não na tabela associada.
Vemos que na definição da tabela aparece como atributo "lógico".

Por outro lado, se o valor do atributo tivéssemos atribuído por meio de uma regra, não deixa de estar presente na tabela pelo simples fato de atribuir-lhe um valor, portanto não se torna um atributo virtual, mas continua sendo um atributo armazenado.

Considerações ao definir um atributo com base em um cálculo: regra ou fórmula?





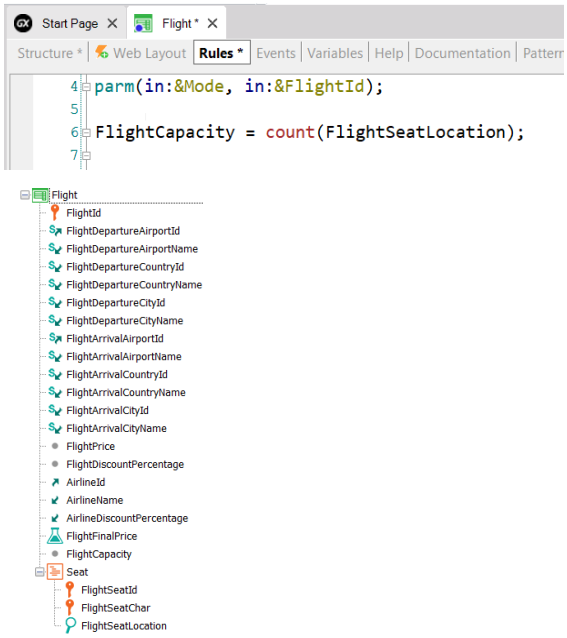
Flight capacity report

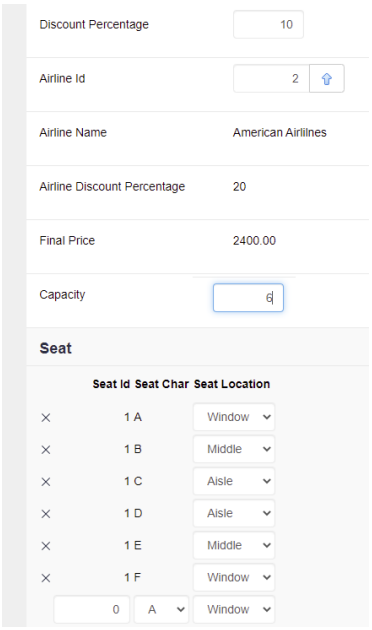
Flight Id	Departure city	Arrival city	Capacity
1	New York	Beijing	12
2	Sao Paulo	Paris	8
3	New York	Sao Paulo	6
4	Paris	Sao Paulo	8

Portanto, é melhor usar uma regra de atribuição do que uma fórmula? Não necessariamente, porque depende do uso que daremos ao atributo.

Se vamos usar o atributo em outros objetos e é necessário ter certeza de que seu valor é o resultado atual do cálculo, então o definimos como fórmula. Como esta definição é global para a base de conhecimento, quando qualquer objeto consulta o valor do atributo, acessa seu cálculo e este é disparado, atualizando o valor no momento.

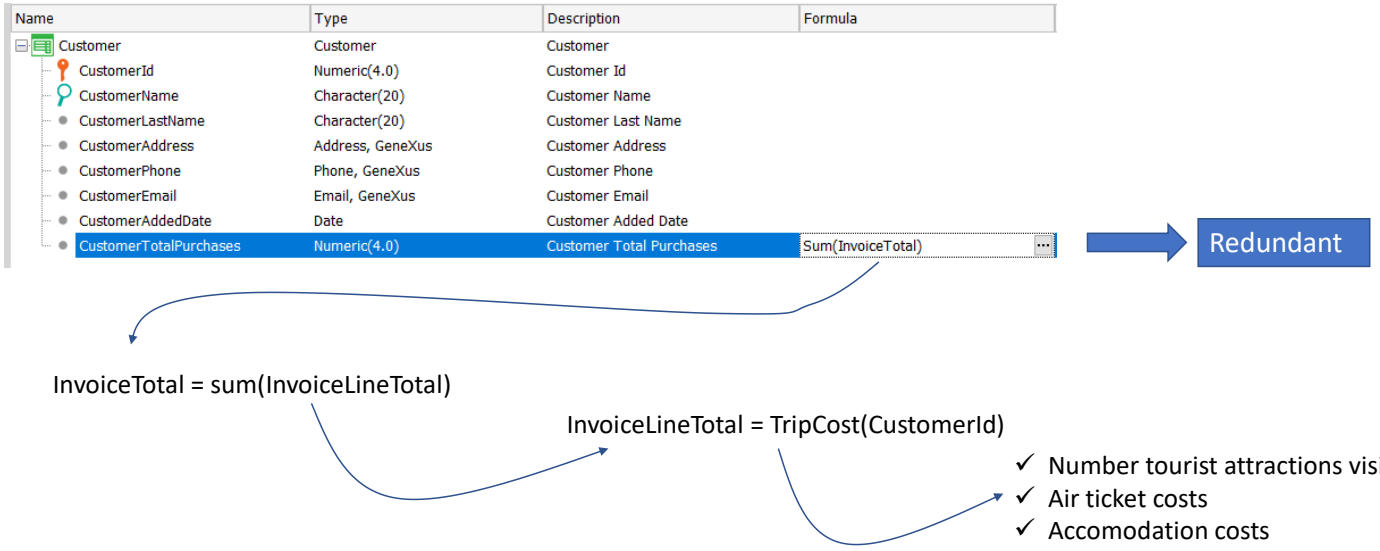
Considerações ao definir um atributo com base em um cálculo: regra ou fórmula?





Se o valor do atributo é atualizado unicamente pela transação, então pode ser atribuído seu valor de forma local, com uma regra. O atributo continua estando armazenado e também pode ser alterado seu valor por meio do form.

Considerações ao definir um atributo com base em um cálculo



Porém, que o atributo deva ser calculado todas as vezes, nem sempre é uma vantagem.

Se o cálculo deve ser realizado sobre muitos registros cada vez e deve ser feito frequentemente, é possível que seja afetado o desempenho da aplicação, em contextos onde são requeridas respostas em tempo real.

Também pode ser o caso quando o atributo fórmula é calculado a partir de outros atributos fórmula, o que significa que deva ser realizada uma grande quantidade de cálculos para obter o valor.

Suponhamos, por exemplo, que estamos calculando o total de compras de um cliente da agência de viagens, como a soma das faturas realizadas ao cliente e o total de cada fatura é calculado como a soma de suas viagens e por sua vez o custo de cada viagem é calculado através de um procedimento que leva em consideração a quantidade de atrações visitadas, o custo das passagens aéreas, o custo das hospedagens, etc.
Se é desejado listar o total de compras de todos os clientes da agência nos últimos 5 anos, certamente deverão ser percorridos muitos registros e serem realizados muitos cálculos, o que pode afetar o tempo de resposta do sistema para obter o resultado.

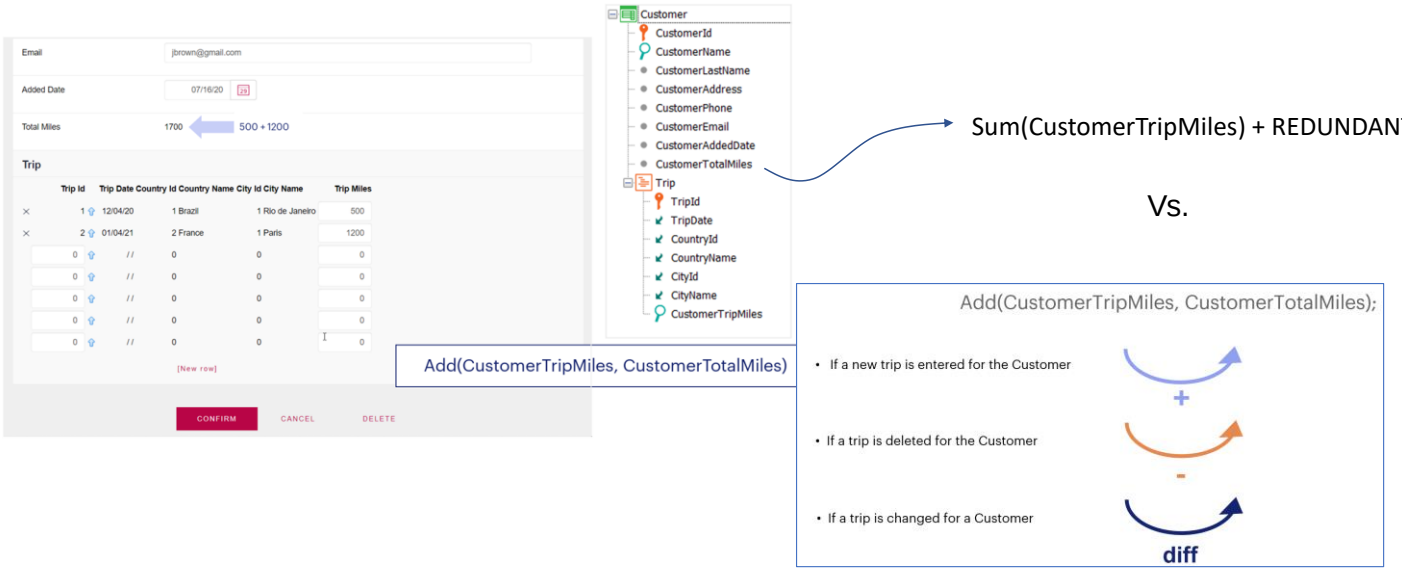
Nesta situação, seria bom que o resultado do total das compras do cliente fosse armazenado em uma tabela, de forma que se algo muda que afete o resultado, seja recalculado novamente e seja armazenado o novo resultado, mas se nada muda, possa ser utilizado o valor armazenado em vez de que se efetue sempre o cálculo.
Fazemos isto definindo o atributo formula como redundante, ou seja, que embora possa ser obtido seu valor por meio de um cálculo, o resultado do mesmo será armazenado na base de dados e o valor será recuperado a

partir dali no futuro.

Também por razões similares, podemos definir um atributo inferido como redundante.

A definição de atributos redundantes veremos posteriormente em outro vídeo.

Atualização de um atributo mediante uma fórmula ou uma regra



Vimos que podemos definir um atributo como fórmula ou atribuir seu valor com uma regra, mas também devemos considerar como será seu mecanismo de atualização em cada caso.

Lembremos o uso da regra Add (ou Subtract), que nos permitia manter atualizado o valor de um atributo da tabela estendida, realizando a operação adequada dependendo se estava sendo inserido, eliminado ou modificado um registro.

O atributo do exemplo CustomerTotalMiles, atualizado pela regra Add, é um atributo armazenado e, portanto, a recuperação do valor é imediata.
No entanto, devido a regra Add ser local para a transação Customer, como vimos anteriormente, somente será atualizado o atributo CustomerTotalMiles se for executada a tela da transação Customer ou um business component da transação.

Se queremos que o valor do atributo do total de milhas do cliente se mantenha sempre atualizado, deveríamos defini-lo como fórmula, neste caso uma fórmula Sum que some as milhas de cada viagem ao total de milhas do cliente.

Embora ao definir o atributo como fórmula asseguremos sua constante atualização, perdemos a possibilidade de que esteja armazenado e para obter seu valor podemos incorrer nos problemas de desempenho que mencionamos. Para resolver isso, poderíamos definir o atributo fórmula CustomerTotalMiles como redundante e passará a ser um atributo armazenado. Mas quando é atualizado o valor do atributo na tabela?

Atualização de um atributo mediante uma fórmula ou uma regra (continuação)

Quando definimos um atributo fórmula como redundante, GeneXus cria automaticamente procedimentos encarregados de atualizar seu valor e armazená-lo na base de dados.

No exemplo que vimos, quando por meio da transação Customer (ou seu Business Component) é inserida ou eliminada uma viagem de um cliente ou é afetado o valor do CustomerTripMiles, a fórmula Sum é recalculada e é armazenado o novo valor na tabela Customer.

Quando a partir do form de uma transação ou a partir de um Business Component se modifica o valor de algum dos atributos que fazem parte do cálculo de um atributo redundante, GeneXus dispara o procedimento que atualiza seu valor.

Portanto, do ponto de vista da atualização e que o atributo esteja armazenado, atualizar o atributo mediante uma regra ADD e defini-lo como fórmula redundante são soluções equivalentes.

Comparação entre uso de uma regra Add e uma fórmula redundante

Vantagens da regra ADD	Desvantagens da regra Add
O atributo atribuído pela regra está sempre armazenado	Não é disparada a regra Add quando é inserido, modificado ou eliminado um registro da tabela através de um objeto procedimento
O atributo é editável no form da transação	Não é possível forçar o disparo da regra sob demanda, portanto, não é possível forçar a atualização do atributo
O atributo é atualizado somente quando é inserido, modificado ou eliminado um registro da transação onde foi definida a regra, através do form ou com BC	
A regra add conhece a operação a ser realizada dependendo do estado da transação (insert, update ou delete)	
O tempo de atualização é mínimo, é acessada a tabela estendida	

No entanto, há várias diferenças que devemos considerar entre usar um mecanismo e o outro.

Vejamos agora algumas tabelas comparativas que nos ajudem a considerar os prós e os contras em cada caso.

Analisemos em primeiro lugar, as vantagens e desvantagens de usar uma regra Add.

Como vantagens temos que:

- O atributo atribuído pela regra Add está sempre armazenado, portanto, a recuperação de seu valor é imediata.
- O atributo continua editável no form da transação
- O atributo é atualizado sempre que é disparada a regra, ou seja, quando é inserido, modificado ou eliminado um registro da transação onde foi definida a regra, seja através de seu form ou por meio de Business Components.
- A regra add conhece a operação a ser realizada dependendo do estado da transação, ou seja, sabe se deve somar quando é inserido um registro, subtrair se é eliminado ou modificar quando é realizado um update.
- O tempo da atualização é mínimo, pois são acessadas somente as tabelas que pertencem à tabela estendida

Como principais desvantagens, vemos que:

- Não é disparada a regra Add quando é inserido, modificado ou eliminado um registro da tabela através de um objeto procedimento, portanto, neste caso o valor do atributo não é atualizado
- Não é possível forçar o disparo da regra sob demanda, portanto, não é possível forçar a atualização do atributo.

Comparação entre uso de uma regra Add e uma fórmula redundante

Vantagens da fórmula redundante	Desvantagens da fórmula redundante
O atributo fórmula global redundante está sempre armazenado	O atributo é read-only no form
O atributo é atualizado quando é modificado um atributo do cálculo, ou quando é inserido, modificado ou eliminado um registro da transação onde foi definido, através do form ou com BC	O tempo de atualização é dispendioso, pois envolve a execução de vários procedimentos e o acesso a várias tabelas
Os procedimentos de redundância conhecem a operação a realizar para manter o atributo atualizado	Aumenta o tamanho da KB devido a serem adicionados os objetos procedimentos criados para manter a redundância
É possível forçar o disparo da atualização da redundância sob demanda, mediante um procedimento especial	Se muda algo da definição da fórmula, GeneXus deve atualizar os procedimentos de redundância

Vejamos agora as vantagens e desvantagens de definir um atributo fórmula como redundante.

Como vantagens podemos citar que:

- O atributo definido como fórmula global redundante está sempre armazenado
- É atualizado quando é modificado qualquer atributo que integra o cálculo, ou quando é inserido, modificado ou eliminado um registro da transação onde foi definido o atributo fórmula, seja através do form ou com Business Components.
- Os procedimentos de redundância conhecem a operação a ser realizada para manter o atributo atualizado, ou seja, sabem se devem somar, subtrair ou como modificar o valor.
- É possível forçar o disparo da atualização da redundância sob demanda, mediante procedimentos especiais que GeneXus cria quando definimos um atributo como redundante.

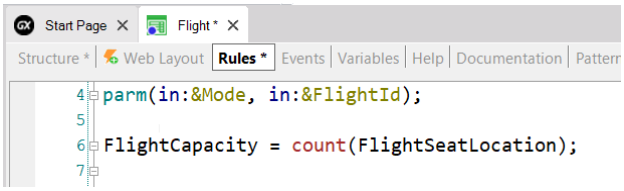
Como desvantagens, temos que:

- O atributo é read-only no form da transação onde foi definido. Por mais que, com a redundância passe a ser armazenado, não podemos editá-lo.
- O tempo de atualização da redundância é dispendioso, pois envolve a execução de vários procedimentos e geralmente o acesso a várias tabelas
- Quando é definido um atributo fórmula como redundante, aumenta

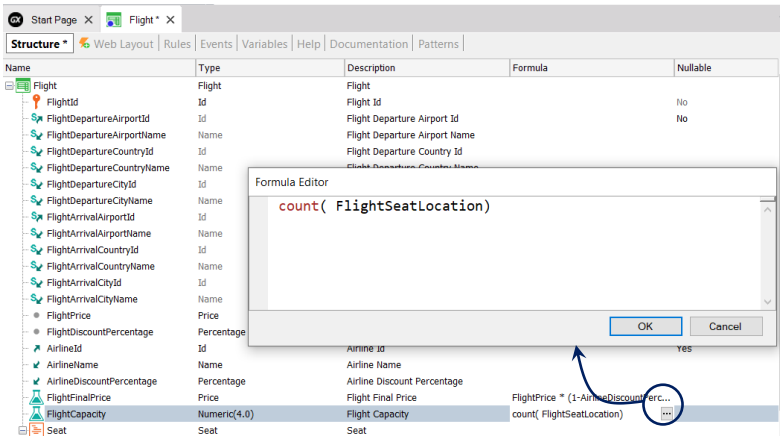
o tamanho da KB devido a serem adicionados os procedimentos criados para manter a redundância

- Se muda algo da definição da fórmula, GeneXus deve manter atualizados os procedimentos de redundância

Considerações ao definir um atributo com base em um cálculo: regra ou fórmula?



Vs.



+ REDUNDANT

Portanto, como dissemos antes, devemos considerar o que é melhor, se atualizar o valor do atributo por meio de uma regra ou definindo-o como fórmula redundante, dependendo das considerações que acabamos de ver e do uso que daremos ao atributo.

Nos vídeos seguintes, estudaremos os diferentes tipos de fórmulas que podemos definir.

GeneXus™

training.genexus.com
wiki.genexus.com