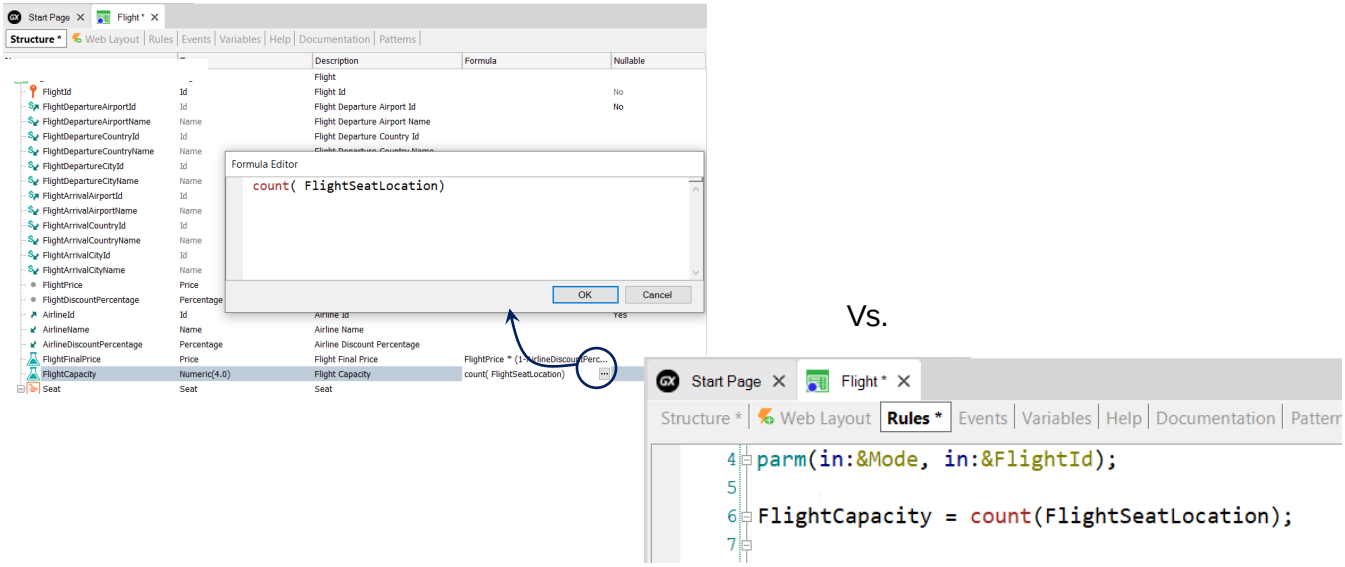
Fórmula vs. regla de asignación

GeneXus

Consideraciones al definir un atributo en base a un cálculo: ¿regla o fórmula?

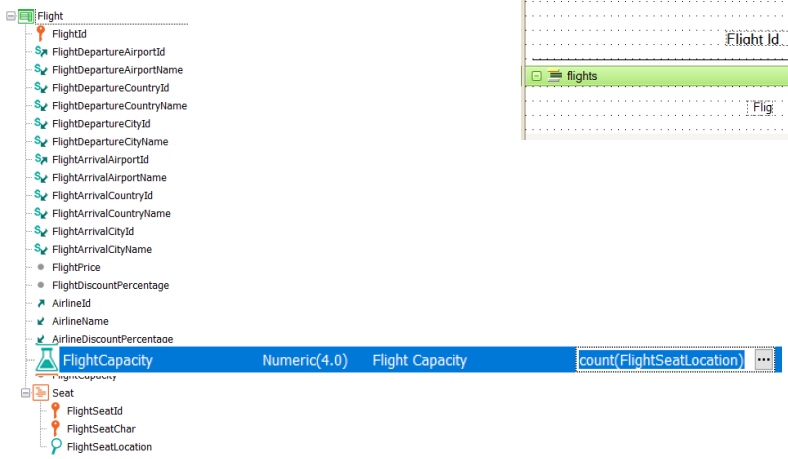


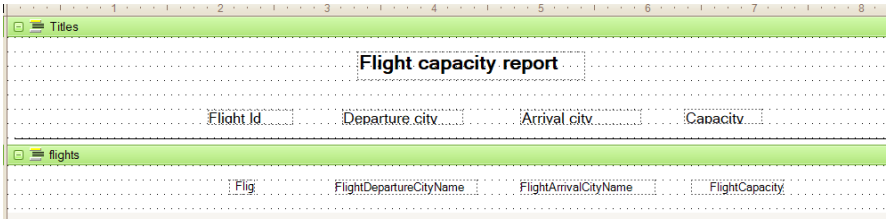
Cuando el valor de un atributo puede obtenerse mediante un cálculo, por lo general lo definimos como fórmula en la estructura de la transacción.
Sin embargo, notemos que también podríamos asignar el cálculo al atributo mediante una regla.

¿Qué consideraciones debemos tomar en cuenta para decidir si asignamos el cálculo mediante una regla o si lo hacemos definiendo el atributo como fórmula? Ya vimos anteriormente que si el atributo es fórmula, GeneXus no crea en su tabla asociada (es decir la tabla a la que pertenecería el atributo fórmula si estuviera almacenado), un campo para almacenar el valor, ya que entiende que su valor puede obtenerse a partir del cálculo. Debido a esto, decimos que consideramos al atributo como "virtual" ya que sigue presente en la estructura de la transacción, pero no en la tabla asociada.
Vemos que en la definición de la tabla aparece como atributo "lógico".

En cambio, si el valor del atributo lo hubiéramos asignado mediante una regla, no deja de estar presente en la tabla por el mero hecho de asignarle un valor, por lo que no se convierte en un atributo virtual, sino que sigue siendo un atributo almacenado.

Consideraciones al definir un atributo en base a un cálculo: ¿regla o fórmula?





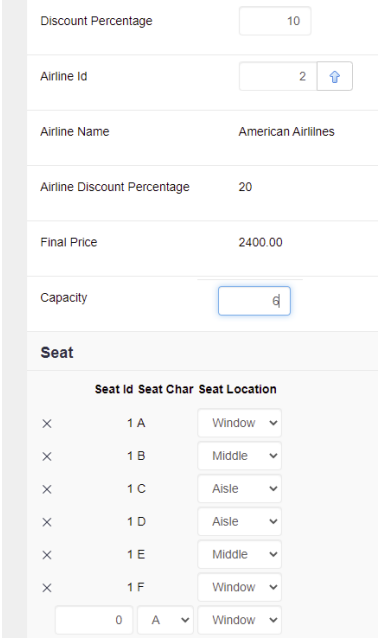
Flight capacity report

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

¿Entonces es mejor usar una regla de asignación que una fórmula? No, necesariamente porque depende del uso que le daremos al atributo.

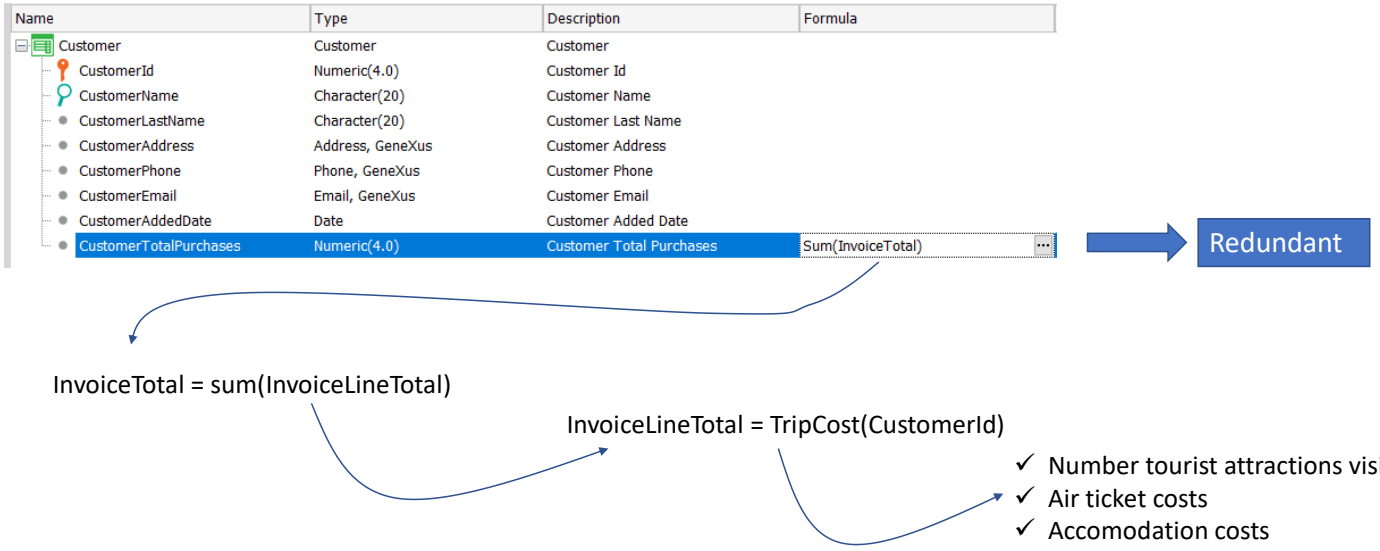
Si al atributo lo vamos a usar en otros objetos y se necesita asegurarse de que su valor es el resultado actual del cálculo, entonces lo definimos como fórmula. Como esta definición es global a la base de conocimientos, cuando cualquier objeto consulta el valor del atributo, accede a su cálculo y éste se dispara, actualizándose el valor en el momento.

Consideraciones al definir un atributo en base a un cálculo: ¿regla o fórmula?



Si el valor del atributo es actualizado únicamente por la transacción, entonces puede asignarse su valor en forma local, con una regla. El atributo sigue estando almacenado y también puede cambiarse su valor a través del form.

Consideraciones al definir un atributo en base a un cálculo



Sin embargo, que el atributo deba calcularse cada vez, no siempre es una ventaja.

Si el cálculo debe realizarse sobre muchos registros cada vez y se tiene que hacer frecuentemente, es posible que se afecte la performance de la aplicación, en contextos donde se requiere repuestas en tiempo real.

También puede ser el caso cuando el atributo fórmula es calculado a partir de otros atributos fórmula, con lo que deba realizarse una gran cantidad de cálculos para obtenerse el valor.

Supongamos por ejemplo que estamos calculando el total de compras de un cliente de la agencia de viajes, como la suma de las facturas realizadas al cliente y el total de cada factura se calcula como la suma de sus viajes y a su vez el costo de cada viaje es calculado mediante un procedimiento que toma en cuenta la cantidad de atracciones visitadas, el costo de los pasajes aéreos, el costo de los alojamientos, etc.
Si se desea listar el total de compras de todos los clientes de la agencia en los últimos 5 años, seguramente deban recorrerse muchos registros y realizarse muchos cálculos, que pueden afectar el tiempo de respuesta del sistema para obtener el resultado.

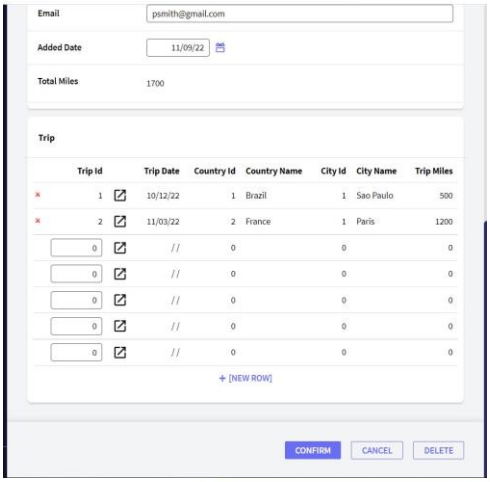
Ante esta situación, sería bueno que el resultado del total de compras del cliente se almacene en una tabla, de modo que si cambia algo que afecte el resultado se recalcule nuevamente y se almacene el nuevo resultado, pero si no cambia nada, se pueda utilizar el valor almacenado en lugar de que se efectúe siempre el cálculo.

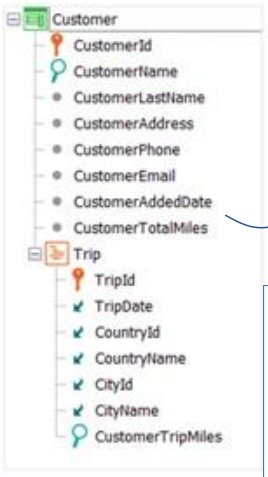
Esto lo hacemos definiendo al atributo fórmula como redundante, es decir, que si bien puede obtenerse su valor mediante un cálculo, el resultado del mismo se almacenará en la base de datos y el valor será recuperado desde allí en el futuro.

También por razones similares, podemos definir a un atributo inferido como redundante.

La definición de atributos redundantes lo veremos más adelante en otro video.

Actualización de un atributo mediante una fórmula o una regla

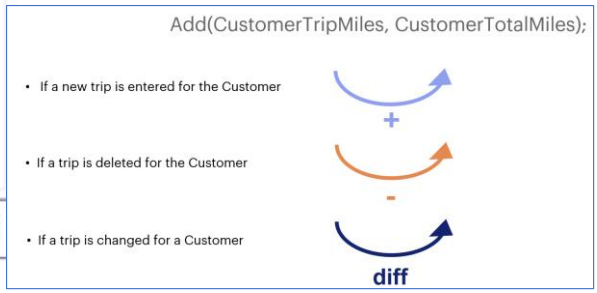




Sum(CustomerTripMiles) + REDUNDANT

Vs.

Add(CustomerTripMiles, CustomerTotalMiles)



Vimos que podemos definir un atributo como fórmula o asignar su valor con una regla pero también debemos considerar cómo será su mecanismo de actualización en cada caso.

Recordemos el uso de la regla Add (o Subtract), que nos permitía mantener actualizado el valor de un atributo de la tabla extendida, realizando la operación adecuada dependiendo de si se estaba insertando, eliminando o modificando un registro.

El atributo del ejemplo CustomerTotalMiles, actualizado por la regla Add, es un atributo almacenado y por lo tanto la recuperación del valor es inmediata.
Sin embargo, debido a que la regla Add es local a la transacción Customer, como vimos antes, solamente se actualizará el atributo CustomerTotalMiles si se ejecuta la pantalla de la transacción Customer o un business component de la transacción.

Si queremos que el valor del atributo del total de millas del cliente se mantenga siempre actualizado, deberíamos definirlo como fórmula, en este caso una fórmula Sum que sume las millas de cada viaje al total de millas del cliente.

Si bien al definir el atributo como fórmula aseguramos su constante actualización, perdemos la posibilidad de que esté almacenado y para obtener su valor podemos incurrir en los problemas de performance que mencionamos. Para solucionarlo, podríamos definir al atributo fórmula CustomerTotalMiles como redundante y pasará a ser un atributo almacenado. ¿Pero cuándo se actualiza el valor del atributo en la tabla?

Actualización de un atributo mediante una fórmula o una regla (continuación)

Cuando definimos un atributo fórmula como redundante, GeneXus crea automáticamente procedimientos encargados de actualizar su valor y almacenarlo en la base de datos.

En el ejemplo que vimos, cuando mediante la transacción Customer (o su Business Component) se inserta o elimina un viaje de un cliente o se afecta el valor del CustomerTripMiles, la fórmula Sum se recalcula y se almacena el nuevo valor en la tabla Customer.

Cuando desde el form de una transacción o desde un Business Component se modifica el valor de algunos de los atributos que forman parte del cálculo de un atributo redundante, GeneXus dispara el procedimiento que actualiza su valor.

Por lo tanto, desde el punto de vista de la actualización y que el atributo esté almacenado, actualizar el atributo mediante una regla ADD y definirlo como fórmula redundante son soluciones equivalentes.

Comparación entre uso de una regla Add y una fórmula redundante

Ventajas de la regla ADD	Desventajas de la regla Add
El atributo asignado por la regla está siempre almacenado	No se dispara la regla Add cuando se inserta, modifica o elimina un registro de la tabla través de un objeto procedimiento
El atributo es editable en el form de la transacción	No se puede forzar el disparo de la regla a demanda, por lo que no se puede forzar la actualización del atributo
El atributo se actualiza sólo cuando se inserta, modifica o elimina un registro de la transacción donde fue definida la regla, a través del form o con BC	
La regla add conoce la operación a realizar dependiendo del estado de la transacción (insert, update o delete)	
El tiempo de la actualización es mínimo, se accede a la tabla extendida	

Si bien desde el punto de vista de la actualización y de que estén almacenados, usar una regla add o definir un atributo fórmula como redundante son soluciones equivalentes, hay varias diferencias que debemos considerar entre usar un mecanismo y el otro.

Veamos ahora unos cuadros comparativos que nos ayude a considerar los pro y los contras en cada caso.

Analicemos en primer lugar las ventajas y desventajas de usar una regla Add.

Como ventajas tenemos que:

- El atributo asignado por la regla Add está siempre almacenado, por lo que la recuperación de su valor es inmediata.
- El atributo sigue siendo editable en el form de la transacción
- El atributo se actualiza cada vez que se dispara la regla, es decir cuando se inserta, modifica o elimina un registro de la transacción donde fue definida la regla, ya sea a través de su form o mediante Business Components.
- La regla add conoce la operación a realizar dependiendo del estado de la transacción, es decir sabe si tiene que sumar cuando se inserta un registro, restar si se elimina o modificar cuando se realiza un update.
- El tiempo de la actualización es mínimo, ya que se accede solamente a tablas que pertenecen a la tabla extendida

Como principales desventas, vemos que:

- No se dispara la regla Add cuando se inserta, modifica o elimina un registro

de la tabla través de un objeto procedimiento, por lo que en este caso el valor del atributo no se actualiza

- No se puede forzar el disparo de la regla a demanda, por lo que no se puede forzar la actualización del atributo.

Comparación entre uso de una regla Add y una fórmula redundante

Ventajas de la fórmula redundante	Desventajas de la fórmula redundante
El atributo fórmula global redudante está siempre almacenado	El atributo es read-only en el form
El atributo se actualiza cuando se modifica un atributo del cálculo, o cuando se inserta, modifica o elimina un registro de la transacción donde fue definido, a través del form o con BC	El tiempo de actualización es costoso, ya que involucra la ejecución de varios procedimientos y el acceso a varias tablas
Los procedimientos de redundancia conocen la operación a realizar para mantener el atributo actualizado	Se incrementa el tamaño de la KB debido a que se agregan los objetos procedimientos creados para mantener la redundancia
Se puede forzar el disparo de la actualización de la redundacias a demanda, mediante un procedimiento especial	Si cambia algo de la definición de la fórmula, GeneXus debe actualizar los procedimientos de redundancia

Veamos ahora las ventajas y desventajas de definir un atributo fórmula como redundante.

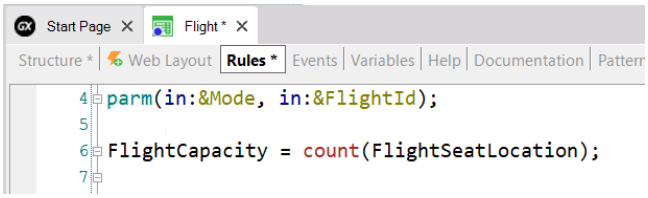
Como ventajas podemos citar que:

- El atributo definido como fórmula global redundante está siempre almacenado
- Se actualiza cuando se modifica cualquier atributo que integra el cálculo, o cuando se inserta, modifica o elimina un registro de la transacción donde fue definido el atributo fórmula, ya sea a través del form o con Business Components.
- Los procedimientos de redundancia conocen la operación a realizar para mantener el atributo actualizado, es decir saben si tienen que sumar, restar o cómo modificar el valor.
- Se puede forzar el disparo de la actualización de la redundacia a demanda, mediante procedimientos especiales que GeneXus crea cuando definimos un atributo como redundante.

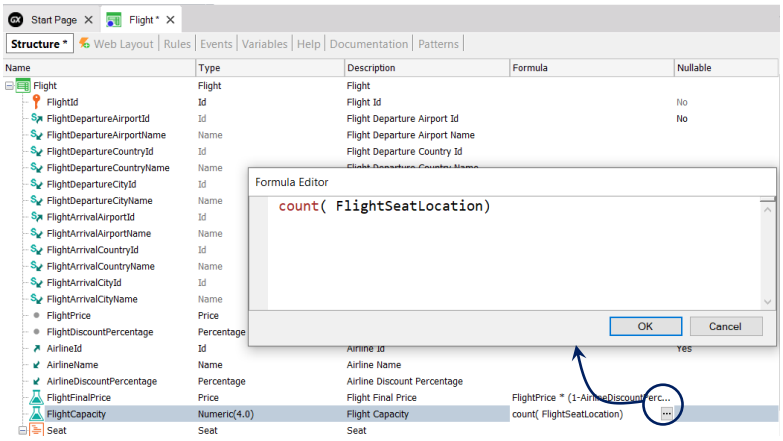
Como desventajas, tenemos que:

- El atributo es read-only en el form de la transacción donde fue definido. Por más que con la redundancia pase a ser almacenado, no lo podemos editar.
- El tiempo de actualización de la redundancia es costoso, ya que involucra la ejecución de varios procedimientos y generalmente el acceso a varias tablas
- Cuando se define un atributo fórmula como redundante se incrementa el tamaño de la KB debido a que se agregan los procedimientos creados para mantener la redundancia
- Si cambia algo de la definición de la fórmula, GeneXus debe mantener actualizados los procedimientos de redundancia

Consideraciones al definir un atributo en base a un cálculo: ¿regla o fórmula?



Vs.



+ REDUNDANT

Por lo tanto, como dijimos antes, debemos considerar qué es mejor, si actualizar el valor del atributo mediante una regla o definiéndolo como fórmula redundante, dependiendo de las consideraciones que acabamos de ver y del uso que le daremos al atributo.

En siguientes videos estudiaremos los distintos tipos de fórmulas que podemos definir.

GeneXus™

training.genexus.com
wiki.genexus.com