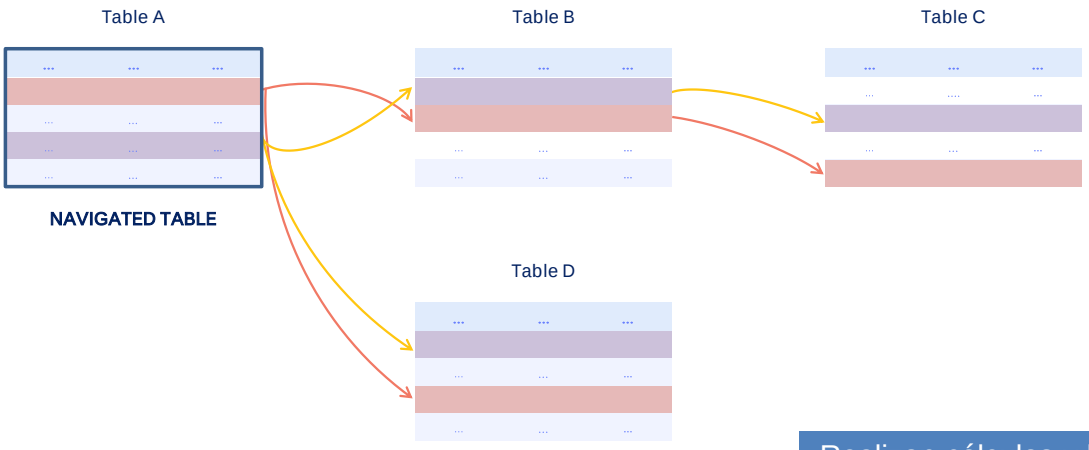
Fórmulas de agregación

GeneXus™

A continuación analizaremos más en detalle las fórmulas aggregate y sus ejemplos de uso.

Repaso de fórmulas aggregate



-Realizan cálculos y búsquedas sobre muchos registros, en cualquier tabla del modelo y su tabla extendida

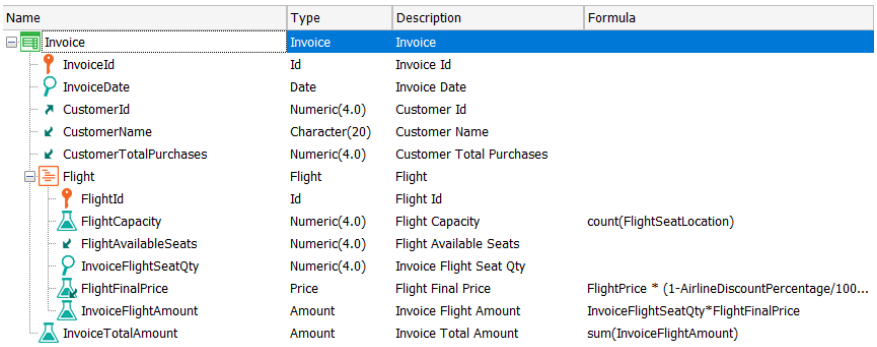
Repasemos primero algunos conceptos.

Las fórmulas aggregate permiten realizar cálculos o búsquedas que involucran muchos registros de una tabla y valores relacionados de la tabla extendida de la tabla recorrida.

Repaso de fórmulas aggregate

Aggregate formulas:

- Count
- Sum
- Average
- Max
- Min
- Find



Name	Type	Description	Formula
Invoice	Invoice	Invoice	
InvoiceId	Id	Invoice Id	
InvoiceDate	Date	Invoice Date	
CustomerId	Numeric(4.0)	Customer Id	
CustomerName	Character(20)	Customer Name	
CustomerTotalPurchases	Numeric(4.0)	Customer Total Purchases	
Flight	Flight	Flight	
FlightId	Id	Flight Id	
FlightCapacity	Numeric(4.0)	Flight Capacity	count(FlightSeatLocation)
FlightAvailableSeats	Numeric(4.0)	Flight Available Seats	
InvoiceFlightSeatQty	Numeric(4.0)	Invoice Flight Seat Qty	
FlightFinalPrice	Price	Flight Final Price	FlightPrice * (1-AirlineDiscountPercentage/100...
InvoiceFlightAmount	Amount	Invoice Flight Amount	InvoiceFlightSeatQty*FlightFinalPrice
InvoiceTotalAmount	Amount	Invoice Total Amount	sum(InvoiceFlightAmount)

Syntax:

AggregateFormula(AggregateExpression, AggregateCondition, DefaultValue, ReturnedValue) if TriggeringCondition

optional optional optional optional

Las fórmulas de agregación no necesitan un contexto, pero si lo hay puede filtrar el resultado

Estas fórmulas nos permiten contar (con Count), sumar (con Sum) o hacer el promedio (con Average) de varios registros, realizar búsquedas como por ejemplo encontrar el valor máximo (con Max) o mínimo (con Min) de un atributo en un conjunto de registros, o dado muchos registros de una tabla, encontrar un valor de un atributo (usando Find) cuyo registro sea el primero encontrado que cumpla determinadas conditiones.

En la sintaxis, la **expresión de agregación** es la expresión que será buscada, maximizada, minimizada, sumada o promediada, entre los registros que cumplen la condición de agregación. Puede contener atributos (incluso atributos fórmula), constantes y variables (las variables creadas por el usuario, sólo se permiten en fórmulas inline). Solo para el caso Count, no es una expresión sino un atributo. Para Sum y Average, el resultado de la expresión de agregación debe ser un valor numérico.

La condición de agregación, es la condición que los registros deben verificar para ser considerados en la agregación. Puede contener atributos, constantes y variables (las de usuario sólo en fórmulas inline). Es opcional y puede no incluirse.

El valor por defecto es el valor que se devolverá cuando no se encuentra ningún registro sobre el que realizar la agregación; puede ser porque ninguno cumple la condición de agregación, o porque no cumple la condición de agregación implícita, o porque la tabla está vacía. Es una constante y también es opcional.

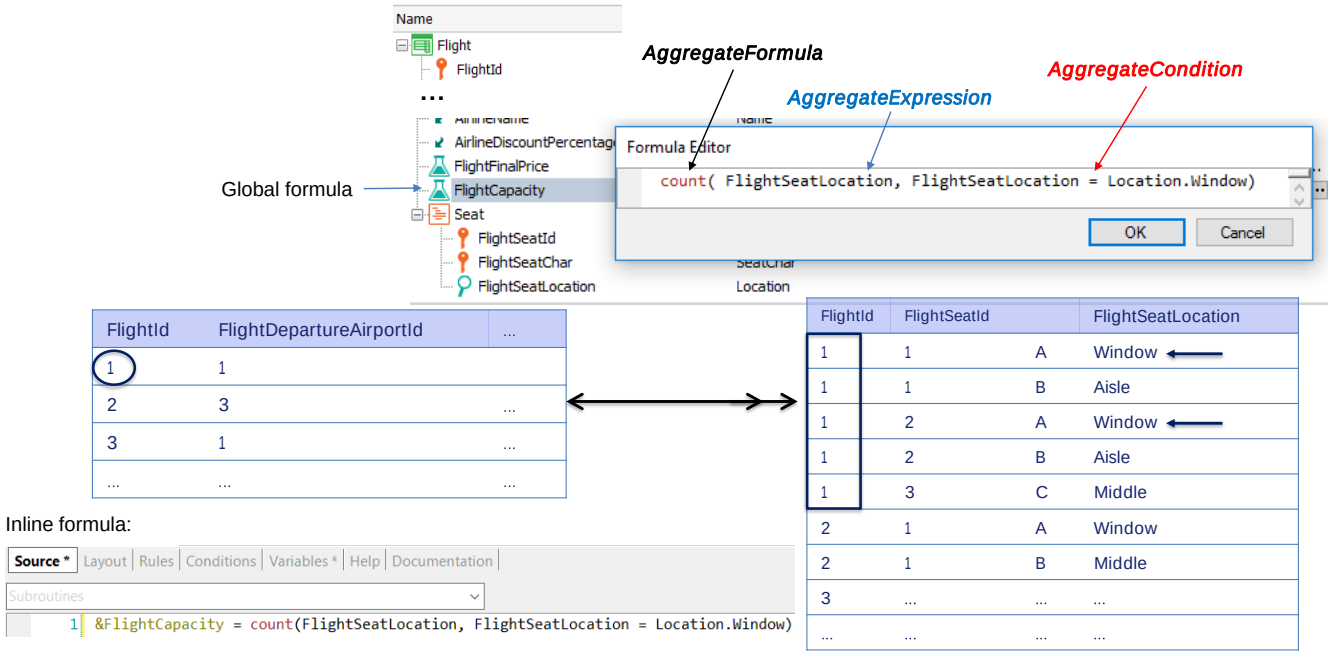
Repaso de fórmulas aggregate (continuación)

El valor retornado, es el atributo cuyo valor es devuelto por la fórmula cuando encuentra registros que cumplen la condición de agregación. Si bien el valor retornado puede ser un valor (como en el caso de que se devuelva el valor por defecto, por lo general, es un atributo. Su inclusión es opcional y solo algunas fórmulas llevan este cuarto parámetro.

La condición de disparo es la que determina que la fórmula se calcule con la expresión de agregación que la antecede.
Es opcional y los únicos atributos permitidos son los que pertenecen a la tabla asociada y a su extendida. Las conditiones de disparo solamente se pueden utilizar en fórmulas aggregate globales ya que en las fórmulas inline no forma parte de la definición de la fórmula, sino que su disparo se condiciona en el código mediante cláusulas condicionales (por ejemplo if, else, do case, etc.) .

Mientras que una fórmula horizontal necesita un contexto para ser evaluada, una fórmula aggregate no necesariamente lo necesita. Sin embargo, si existe una tabla de contexto, no se considerarán todos los registros que la fórmula establece explícitamente, sino solo aquellos que también coincidan con las condiciones implícitas que surgen de ese contexto.

Ejemplo: Count



Inline formula:

Por ejemplo el atributo FlightCapacity de la transacción Flight, es una fórmula count que recorre la tabla FLIGHTSEAT y cuenta la cantidad de asientos del vuelo.

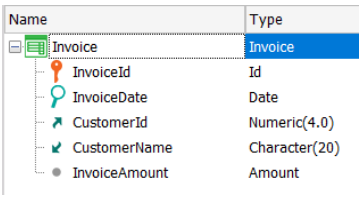
Como la tabla asociada al atributo es la tabla, Flight, que tiene una relación de 1 a N con la tabla FLIGHTSEAT navegada por el Count, se contarán únicamente los registros relacionados, es decir los asientos correspondientes al vuelo instanciado en Flight. Si no hubiera relación, se contarían todos los registros del tabla FLIGHTSEAT.

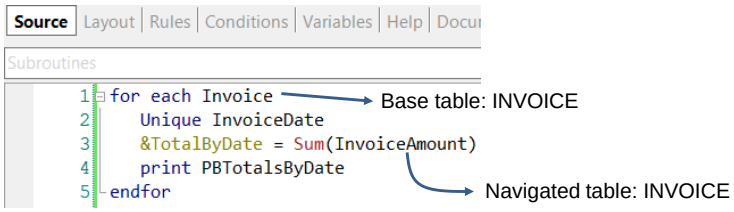
Además como indicamos que solamente se cuenten los asientos que sean ventana, del conjunto de los registros relacionados se contarán únicamente los que cumplan la condición de filtro.

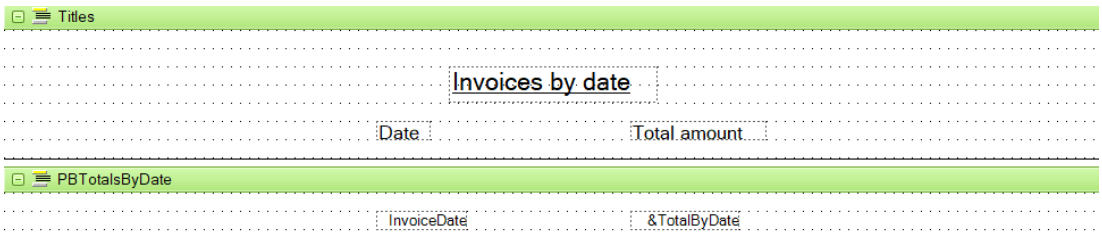
Vemos que como es una fórmula Count, la expresión de agregación es únicamente un atributo que determina la tabla donde se contarán los registros, en este caso FlightSeatLocation, la condición de agregación es el filtro que deben cumplir los asientos (que sean ventana), como no es una fórmula Max o Min no tiene valor por defecto, ni valor retornado y tampoco hay condición de disparo definida.

Aquí vemos a la misma definición como fórmula inline en el source de un procedimiento. En este caso donde la fórmula se encuentra "suelta" no hay condición implícita: se contarán los asientos ventana de todos los vuelos, no de un vuelo en particular, como era el caso de la fórmula global" o como sería el caso si se colocara dentro de este for each

Ejemplo : tabla asociada = tabla navegada







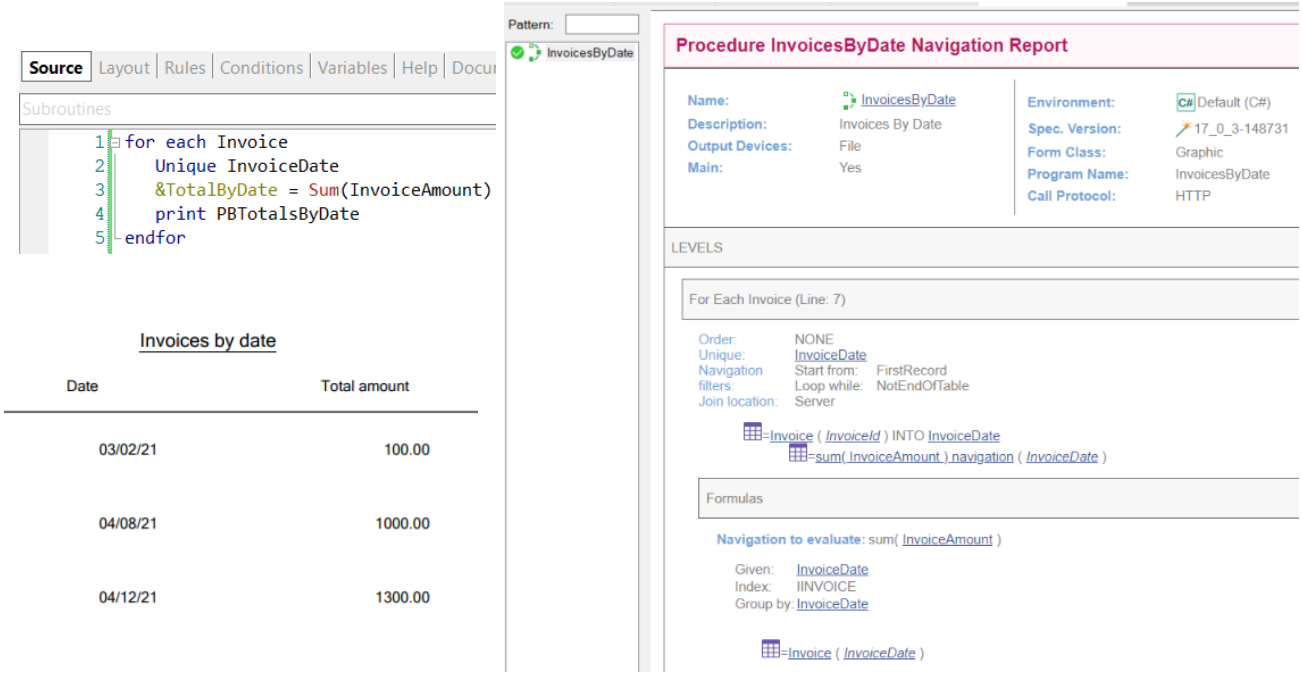
Analicemos un caso particular que es cuando la fórmula está en un cierto contexto y la tabla navegada por la fórmula coincide con esa tabla de contexto.

En este ejemplo queremos imprimir para cada fecha de factura, la suma total de todas las facturas con esa fecha de factura.

Vemos que la tabla navegada por la fórmula Sum es Invoice, y coincide con la tabla base del For Each, que también es Invoice.

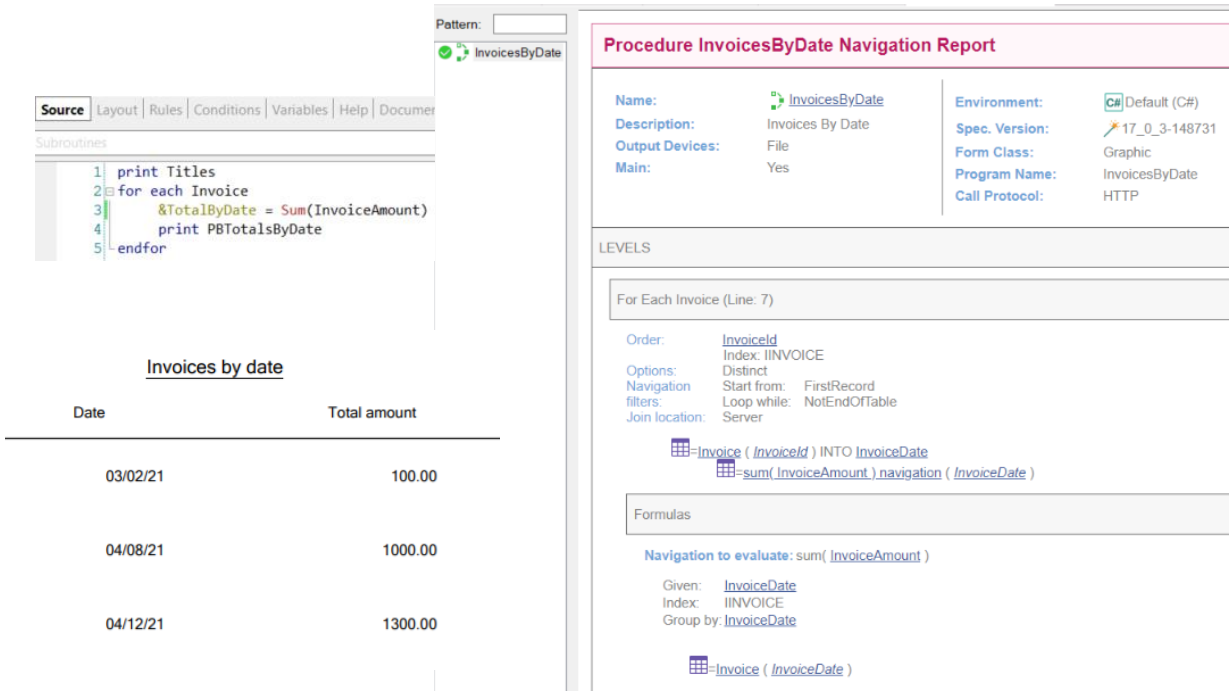
En este caso, debido a la cláusula Unique por InvoiceDate, GeneXus agrupará la información por la fecha de la factura, tanto en el For Each como en la fórmula Sum. Es decir, la fórmula Sum tendrá una condición implícita por igualdad por parte del atributo InvoiceDate, que se considerará dado.

Es como si fuera un corte de control, cortando por InvoiceDate. Veamos el listado de navegación.



Comprobamos que la tabla base del for each es Invoice y que el Unique es por InvoiceDate.

Más abajo vemos la fórmula que también está agrupando por InvoiceDate (observemos el Group by) y que además InvoiceDate es Given. Por tanto va a sumar únicamente los registros de **esa** fecha, tal como queríamos



¿Qué pasaría si quitamos la cláusula Unique?
Si no colocamos el Unique, cómo sabe GeneXus que queremos sumar únicamente los totales de facturas de la fecha de factura en la que se está posicionado?
Es decir, si en vez de escribir la fórmula la descomponemos en su cálculo a través de un for each...

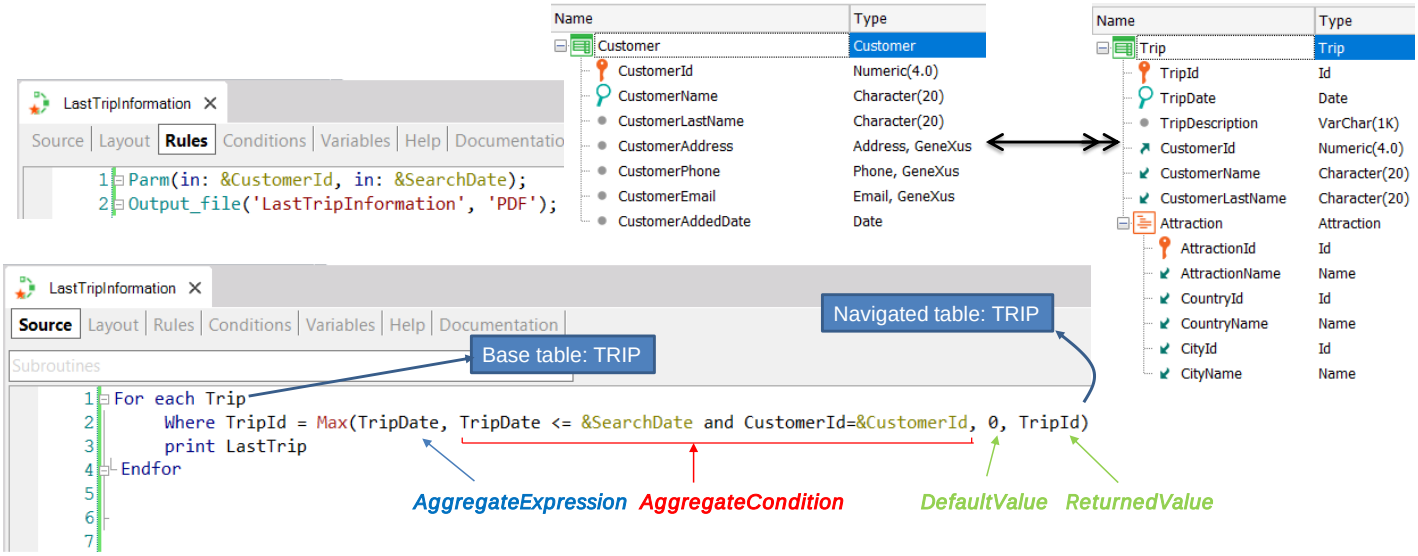
¿Qué saldrá impreso? Tenemos un corte de control, sí, pero... no estamos indicando el criterio de corte por InvoiceDate. No hemos especificado cláusula order para el for each exterior, por lo que elegirá clave primaria, por lo que siempre se quedará en el for each interno con un único registro, el de InvoiceId.

Sin embargo, vamos a ver el listado de navegación que nos muestra GeneXus.

Observemos que no aparece más la cláusula Unique, pero GeneXus agrega automáticamente una cláusula DISTINCT, por lo que el resultado sigue siendo el mismo. GeneXus tuvo la inteligencia de detectar que se deseaba agrupar por fecha de factura y reflejó eso en la generación del código, haciendo que el listado funcionara como nosotros queríamos.

Si bien GeneXus, dependiendo del caso, es capaz de detectar automáticamente lo que desea hacer el desarrollador, lo correcto es que seamos nosotros los que incluyamos la cláusula Unique, para que quede clara nuestra intención en la programación del código.

Ejemplo 2: tabla asociada = tabla navegada



Veamos ahora un caso similar donde se navega una tabla y se realiza una agregación sobre la misma tabla. La agregación será en este caso una fórmula max.

Supongamos que dado un cliente que tiene muchos viajes, quiere recuperar los datos de un viaje realizado, que haya sido inmediatamente anterior a una fecha dada, es decir el último viaje antes de esa fecha.

Supongamos que se desea que la información salga en un listado que imprima los datos del viaje. El procedimiento recibe por parámetro el identificador del cliente que desea la información y la fecha de búsqueda.

En el source del procedimiento hay un for each que recorre la tabla Trip y el where filtrará por el TripId devuelto por la fórmula Max. Luego se imprimirán los datos correspondientes a ese viaje.

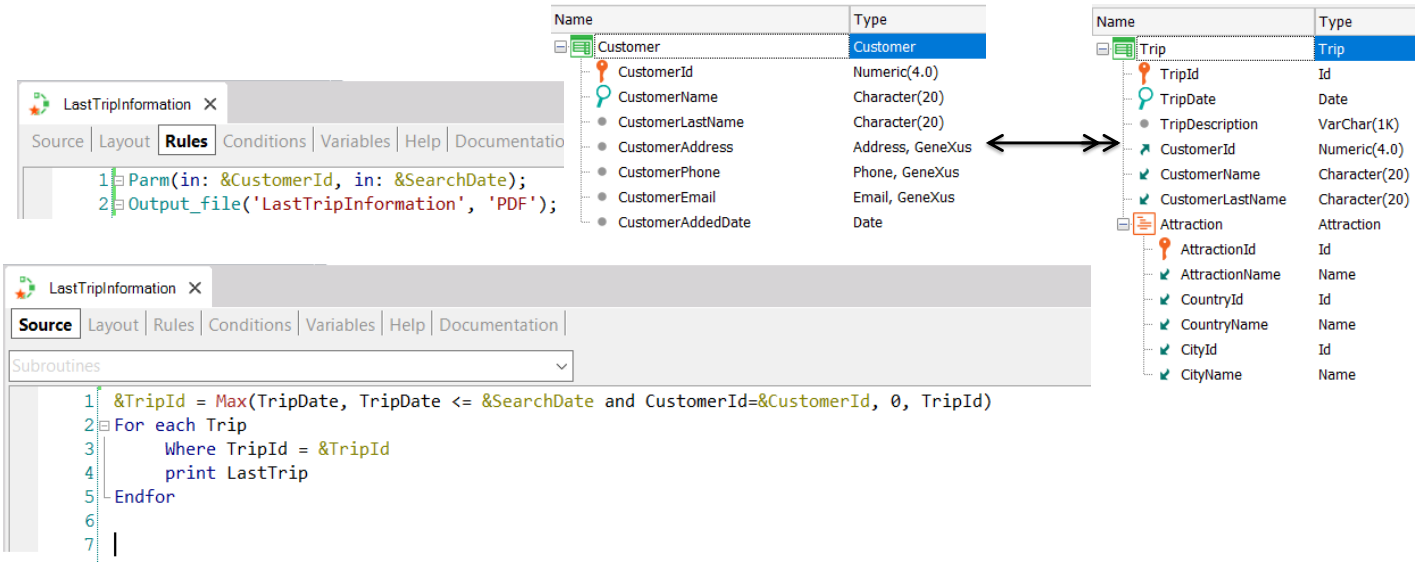
En la fórmula Max, el atributo a maximizar es TripDate, ya que queremos el viaje que haya sido inmediatamente anterior a una fecha dada, por lo tanto será el viaje con la mayor fecha posible, pero menor o igual a la fecha de búsqueda.
La condición de agregación especifica que el viaje debe tener una fecha menor o igual a la fecha dada y que además sea del cliente que busca la información. El valor por defecto devuelto si no se encuentra ningún viaje que cumpla con esas condiciones es el 0 y en caso de que se encuentre será el identificador del viaje encontrado.

Ahora analicemos lo que hemos programado. El For Each iterará sobre la tabla TRIP y establecerá ese contexto para la fórmula Max.
La fórmula Max también iterará en la tabla TRIP, pero debido al contexto, se establecerá un

filtro automático, debido a que ambas tablas están relacionadas. En este caso es la misma tabla, por lo que la fórmula quedará filtrada por el TripId que esté posicionado el For Each. Por tanto va a devolver siempre ese TripId en el que el for each se encuentra, y es como no haber escrito nada. O sea, no nos sirve.

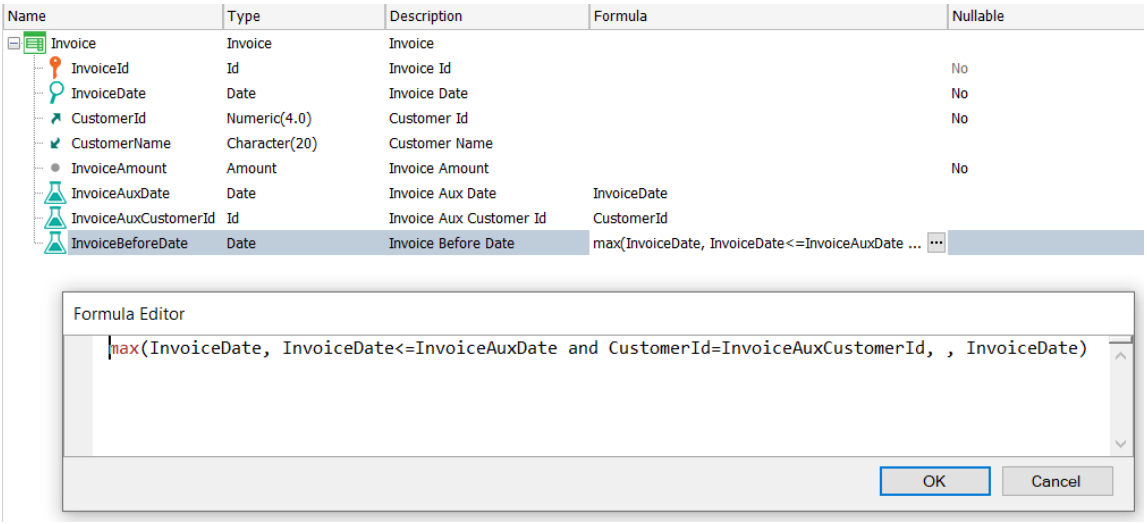
En definitiva, tenemos que tener cuidado cuando una fórmula agreggate navega la misma tabla que la de su contexto, porque la mayoría de las veces, si no explicitamos lo contrario, aplicará un filtro por clave primaria, que tendrá como efecto que la tabla navegada no sea navegada en absoluto y siempre recupere el mismo registro en el que se estaba posicionado

Ejemplo 2: tabla asociada = tabla navegada



La solución en este caso, es ejecutar primero la fórmula para busque el identificador del viaje que cumple las condiciones deseadas y luego filtramos al For Each por ese valor de identificador.

Ejemplo 3: tabla asociada = tabla navegada



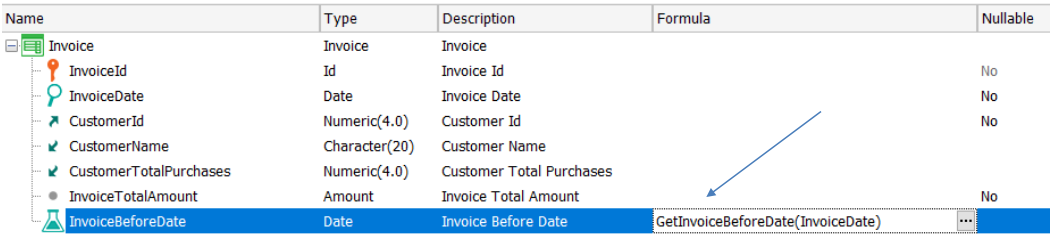
Name	Type	Description	Formula	Nullable
Invoice	Invoice	Invoice		
InvoiceId	Id	Invoice Id		No
InvoiceDate	Date	Invoice Date		No
CustomerId	Numeric(4.0)	Customer Id		No
CustomerName	Character(20)	Customer Name		
InvoiceAmount	Amount	Invoice Amount		No
InvoiceAuxDate	Date	Invoice Aux Date	InvoiceDate	
InvoiceAuxCustomerId	Id	Invoice Aux Customer Id	CustomerId	
InvoiceBeforeDate	Date	Invoice Before Date	max(InvoiceDate, InvoiceDate<=InvoiceAuxDate ...	

Ahora veamos un caso similar al anterior en una fórmula global en el que dada una factura de un cliente, se desea encontrar la fecha de la factura anterior del mismo cliente. Para resolver esto utilizaremos una fórmula Max similar a la que definimos antes, pero definiendo un atributo como fórmula global.

Notemos que otra vez estamos intentando definir una fórmula que navegará sobre la misma tabla asociada a la fórmula, ´por lo que otra vez la fórmula quedará filtrada por su contexto y solamente navegará por un único registro!

la diferencia entre este caso que es una fórmula global y el anterior que era una fórmula inline, es que en el anterior podíamos disparar la fórmula por fuera de la tabla de contexto para romper ese filtro implícito, pero que en el caso de una fórmula global esto no es posible, por lo que NUNCA podremos tener una fórmula aggregate global que navegue la misma tabla.

Ejemplo 3: tabla asociada = tabla navegada



Name	Type	Description	Formula	Nullable
Invoice	Invoice	Invoice		
InvoiceId	Id	Invoice Id		No
InvoiceDate	Date	Invoice Date		No
CustomerId	Numeric(4.0)	Customer Id		No
CustomerName	Character(20)	Customer Name		
CustomerTotalPurchases	Numeric(4.0)	Customer Total Purchases		
InvoiceTotalAmount	Amount	Invoice Total Amount		No
InvoiceBeforeDate	Date	Invoice Before Date	GetInvoiceBeforeDate(InvoiceDate)	

Las fórmulas de agregación no necesitan un contexto, pero si lo hay puede filtrar el resultado

Para ello tendremos, en todo caso, que definirla como fórmula horizontal y realizar el cálculo dentro de un procedimiento.

En resumen, esto verifica lo que dijimos antes, de que las fórmulas aggregate si bien no necesitan un contexto para ser evaluadas (como sí lo necesitan las horizontales), en caso de que ese contexto exista, no se considerarán todos los registros que la fórmula establece explícitamente, sino solo aquellos que también coincidan con las condiciones implícitas que surgen de ese contexto.

Y es importante verificar que el contexto no nos invalide el objetivo que buscábamos con la fórmula, como vimos en los últimos ejemplos presentados.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications