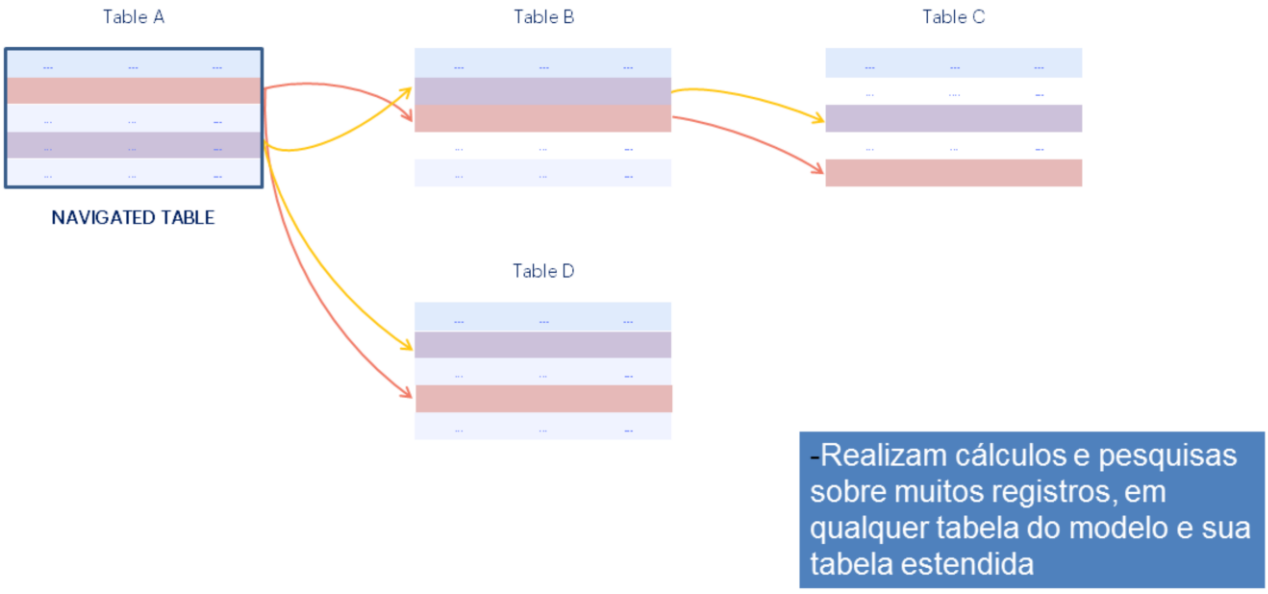
Fórmulas de agregação

GeneXus

A seguir iremos analisar mais detalhadamente as fórmulas aggregate e seus exemplos de uso.

Revisão de fórmulas aggregate



Revisemos primeiro alguns conceitos.

As fórmulas aggregate permitem realizar cálculos ou pesquisas que envolvem muitos registros de uma tabela e valores relacionados da tabela estendida da tabela percorrida.

Revisão 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

As fórmulas de agregação não precisam de um contexto, mas se houver, pode filtrar o resultado

Estas fórmulas nos permitem contar (com Count), somar (com Sum) ou fazer a média (com Average) de vários registros, realizar pesquisas como por exemplo encontrar o valor máximo (com Max) ou mínimo (com Min) de um atributo em um conjunto de registros, ou dados muitos registros de uma tabela, encontrar um valor de um atributo (usando Find) cujo registro seja o primeiro encontrado que atenda a determinadas condições.

Na sintaxe, a **expressão de agregação** é a expressão que será buscada, maximizada, minimizada, somada ou calculada a média, entre os registros que atendem à condição de agregação. Pode conter atributos (inclusive atributos fórmula), constantes e variáveis (as variáveis criadas pelo usuário só são permitidas em fórmulas inline).

Em fórmulas Globais só podem ser utilizadas as variáveis que existem em todo o objeto GeneXus, ou seja, as de sistema, como a variável &Today, por exemplo.

Apenas para o caso de Count, não é uma expressão, mas um atributo. Para Sum e Average, deve ser um valor numérico.

A condição de agregação é a condição que os registros devem verificar para serem considerados na agregação. Pode conter atributos, constantes e variáveis (as de usuário apenas em fórmulas inline). É opcional e pode não ser incluída.

O valor padrão é o valor que será retornado quando não for encontrado nenhum registro sobre o qual realizar a agregação; pode ser porque nenhum atende à condição de agregação, ou porque não atende à condição de agregação implícita, ou porque a tabela está vazia. É uma

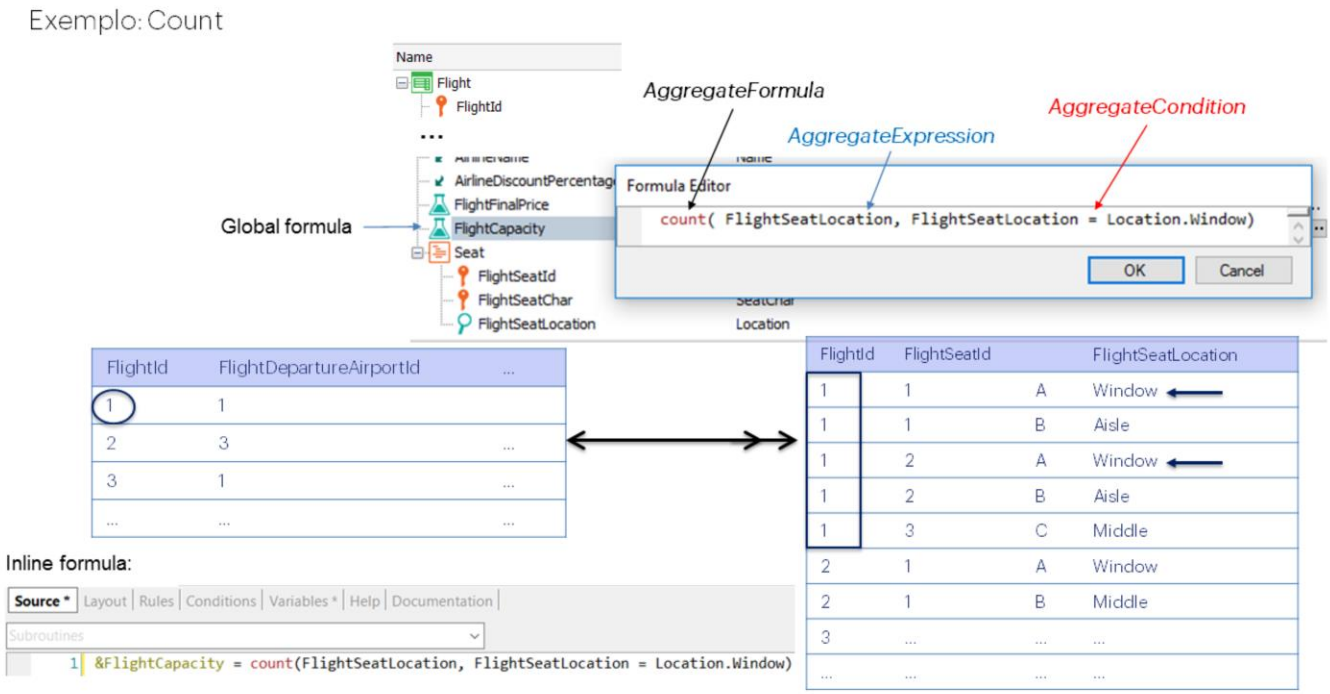
constante e também é opcional.

Revisão de fórmulas aggregate (continuação)

O valor retornado é o atributo cujo valor é retornado pela fórmula quando encontra registros que atendem à condição de agregação. Embora o valor retornado possa ser um valor (como no caso em que seja retornado o valor padrão, geralmente é um atributo. Sua inclusão é opcional e apenas algumas fórmulas possuem este quarto parâmetro.

A condição de disparo é a que determina que a fórmula seja calculada com a expressão de agregação que a precede.
É opcional e os únicos atributos permitidos são aqueles que pertencem à tabela associada e à sua estendida. As condições de disparo só podem ser utilizadas em fórmulas aggregate globais, pois nas fórmulas inline não fazem parte da definição da fórmula, mas seu disparo é condicionado no código mediante cláusulas condicionais (por exemplo, if, else, do case, etc.).

Enquanto uma fórmula horizontal precisa de um contexto para ser avaliada, uma fórmula aggregate não precisa necessariamente dele. No entanto, se existir uma tabela de contexto, não serão considerados todos os registros que a fórmula define explicitamente, mas apenas aqueles que também correspondam com as condições implícitas que surgem desse contexto.



Por exemplo, o atributo FlightCapacity da transação Flight é uma fórmula count que percorre a tabela FLIGHTSEAT e conta a quantidade de assentos do voo.

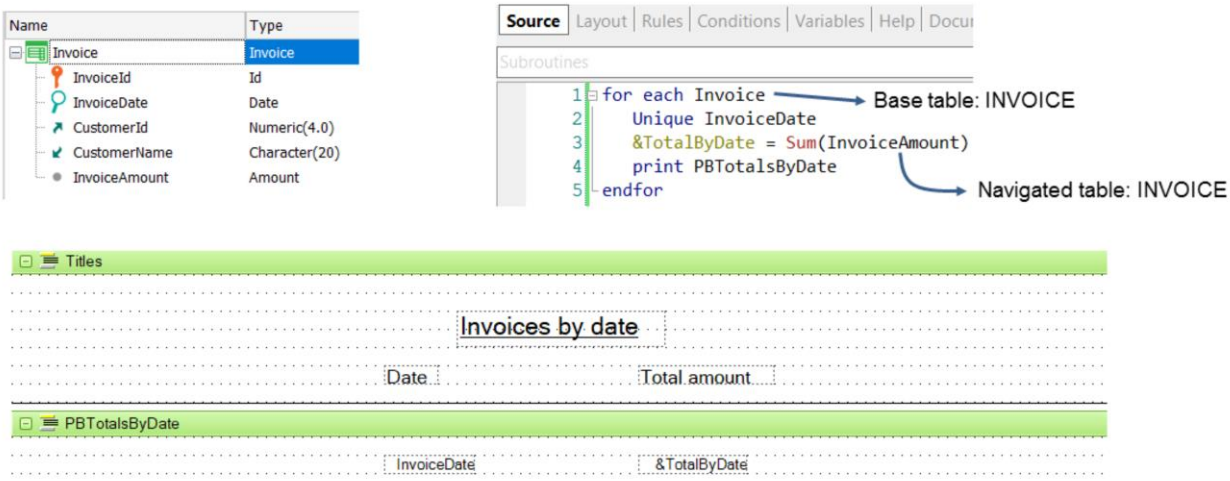
Como a tabela associada ao atributo é a tabela Flight, que tem uma relação de 1 para N com a tabela FLIGHTSEAT navegada pelo Count, serão contados unicamente os registros relacionados, ou seja, os assentos correspondentes ao voo instanciado em Flight. Se não houvesse relação, seriam contados todos os registros da tabela FLIGHTSEAT.

Além disso, como indicamos que apenas sejam contados os assentos que sejam janela, do conjunto dos registros relacionados, serão contados unicamente aqueles que atendem à condição de filtro.

Vemos que por ser uma fórmula Count, a expressão de agregação é unicamente um atributo que determina a tabela de onde serão contados os registros, neste caso FlightSeat, a condição de agregação é o filtro que devem cumprir os assentos (que sejam janela), como não é uma fórmula Max ou Min, não tem valor padrão, nem valor retornado e não há condição de disparo definida.

Aqui, vemos a mesma definição como fórmula inline no source de um procedimento. Neste caso em que a fórmula se encontra "solta" não há condição implícita: serão contados os assentos janela de todos os voos, não de um determinado voo, como era o caso da fórmula global" ou como seria o caso se fosse colocada dentro deste for each

Exemplo : tabela associada = tabela navegada



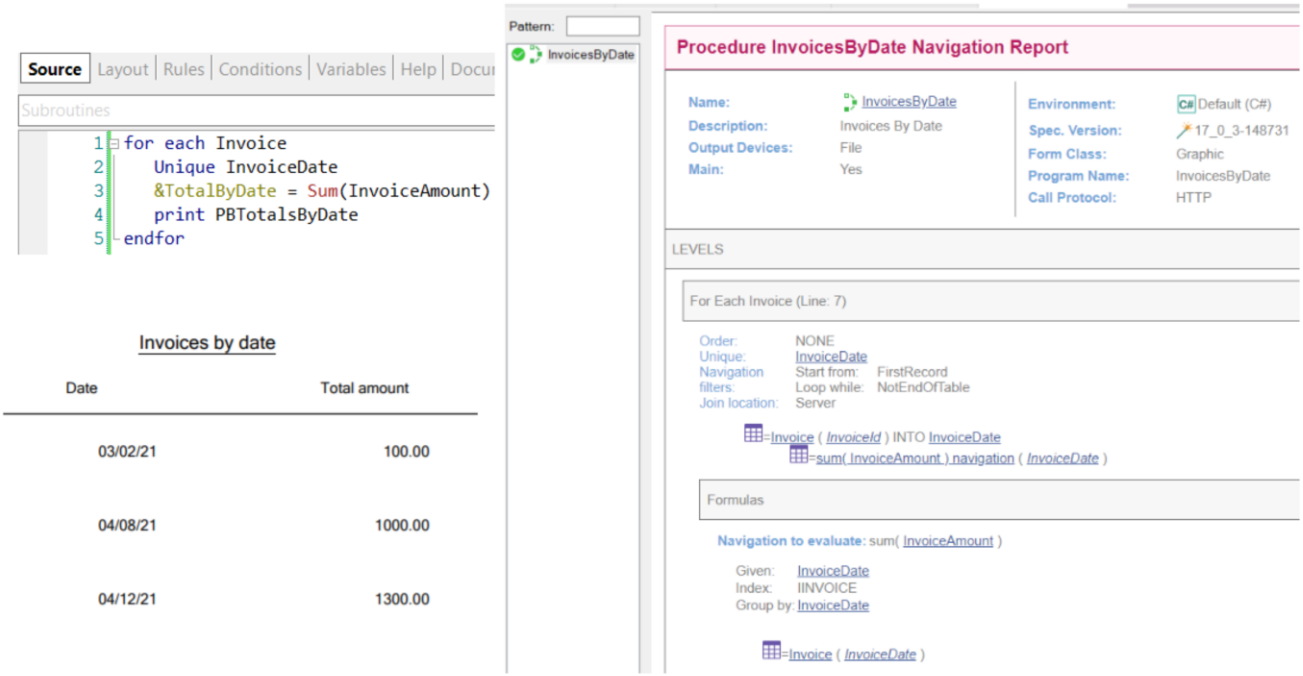
Vamos analisar um caso particular que é quando a fórmula está em um determinado contexto e a tabela navegada pela fórmula coincide com essa tabela de contexto.

Neste exemplo, queremos imprimir para cada data de fatura, a soma total de todas as faturas com essa data de fatura.

Vemos que a tabela navegada pela fórmula Sum é Invoice e coincide com a tabela base do For Each, que também é Invoice.

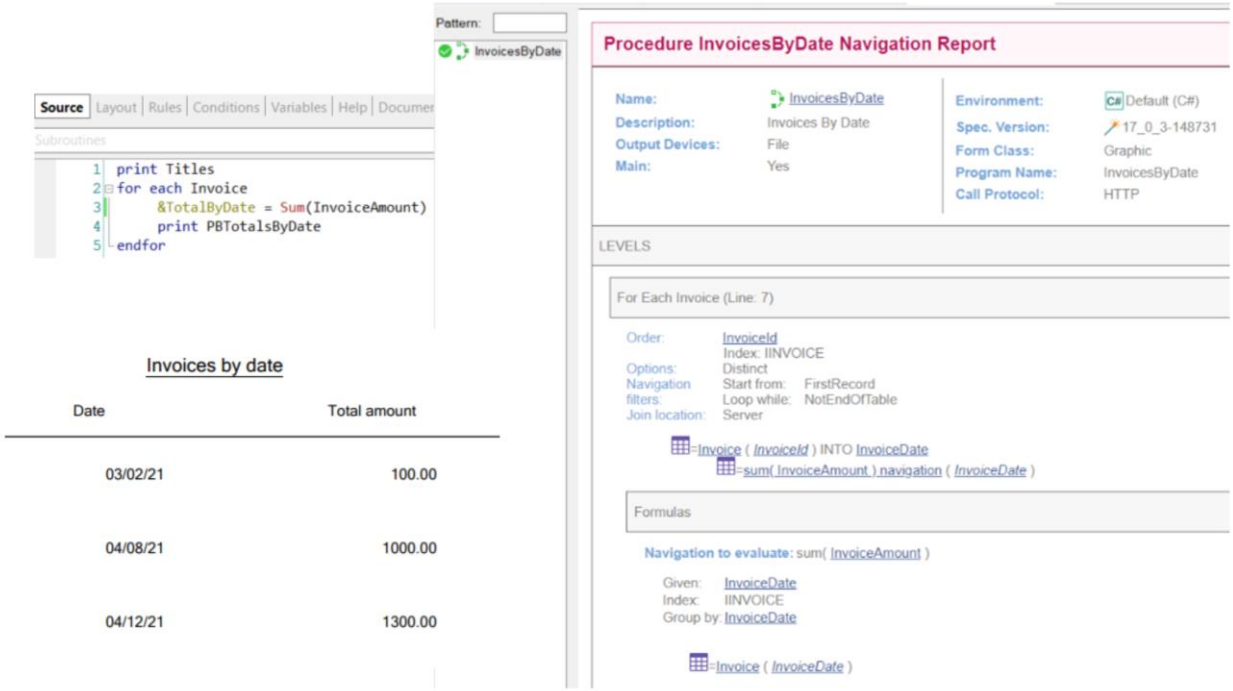
Neste caso, devido à cláusula Unique por InvoiceDate, GeneXus agrupará a informação pela data da fatura, tanto no For Each como na fórmula Sum. Ou seja, a fórmula Sum terá uma condição implícita por igualdade por parte do atributo InvoiceDate, que será considerado dado.

É como se fosse um corte de controle, cortando por InvoiceDate. Vejamos a lista de navegação.



Comprovamos que a tabela base do for each é Invoice e que o Unique é por InvoiceDate.

Mais abaixo, vemos a fórmula que também está agrupando por InvoiceDate (observemos o Group by) e que também InvoiceDate é Given. Portanto, irá somar unicamente os registros **dessa** data, como queríamos



O que aconteceria se removêssemos a cláusula Unique?
Se não colocamos o Unique, como GeneXus sabe que queremos somar apenas os totais de faturas da data de fatura na qual se está posicionado? Ou seja, se em vez de escrever a fórmula a decompormos em seu cálculo através de um for each...

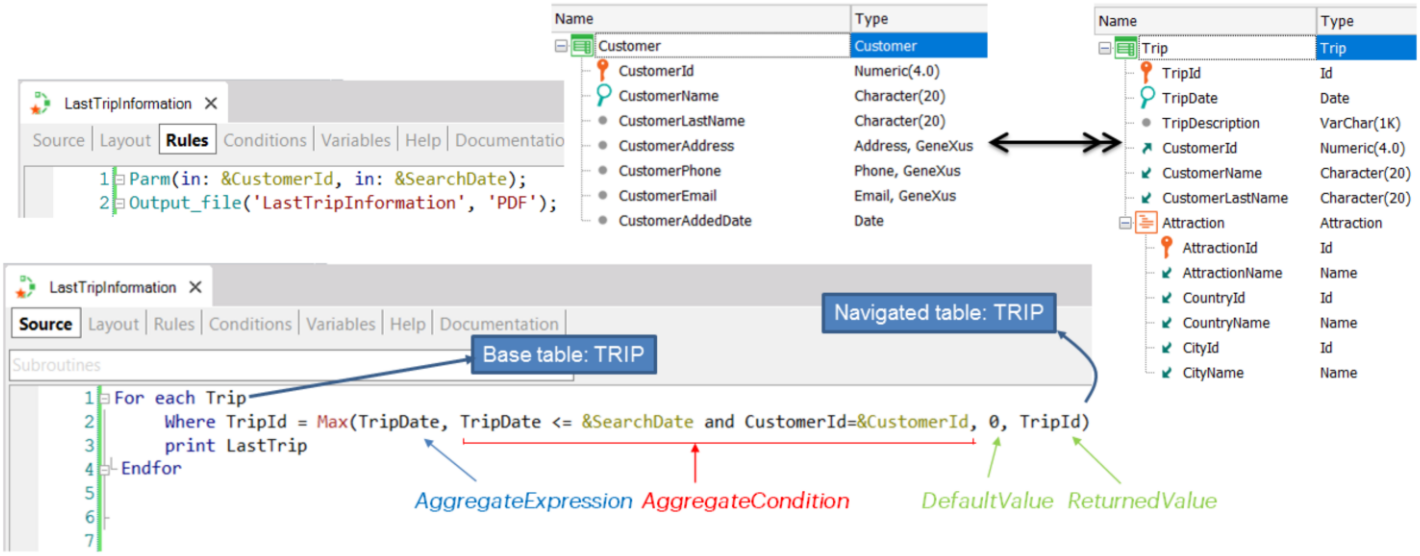
O que será impresso? Temos um corte de controle, sim, mas... não estamos indicando o critério de corte por InvoiceDate. Não especificamos cláusula order para o for each externo, então ele escolherá a chave primária, portanto, sempre se ficará no for each interno com um único registro, o do InvoiceId.

Mesmo assim, vamos ver a lista de navegação que GeneXus nos mostra.

Observemos que não aparece mais a cláusula Unique, mas GeneXus adiciona automaticamente uma cláusula DISTINCT, então o resultado permanece o mesmo. GeneXus teve a inteligência de detectar que se desejava agrupar por data de fatura e refletiu isso na geração do código, fazendo com que a lista funcionasse como queríamos.

Embora GeneXus, dependendo do caso, seja capaz de detectar automaticamente o que deseja fazer o desenvolvedor, o correto é que sejamos nós que incluímos a cláusula Unique, para que fique clara nossa intenção na programação do código.

Exemplo 2: tabela associada = tabela navegada



Vejamos agora um caso semelhante em que é navegada uma tabela e é realizada uma agregação sobre a mesma tabela. A agregação será, neste caso, uma fórmula max.

Suponhamos que dado um cliente que tem muitas viagens, deseja recuperar os dados de uma viagem realizada, que tenha sido imediatamente anterior a uma data dada, ou seja, a última viagem antes dessa data.

Suponhamos que é desejado que as informações apareçam em uma lista que imprima os dados da viagem. O procedimento recebe por parâmetro o identificador do cliente que deseja a informação e a data de busca.

No source do procedimento existe um for each que percorre a tabela Trip e o where filtrará pelo TripId retornado pela fórmula Max. Em seguida, serão impressos os dados correspondentes a essa viagem.

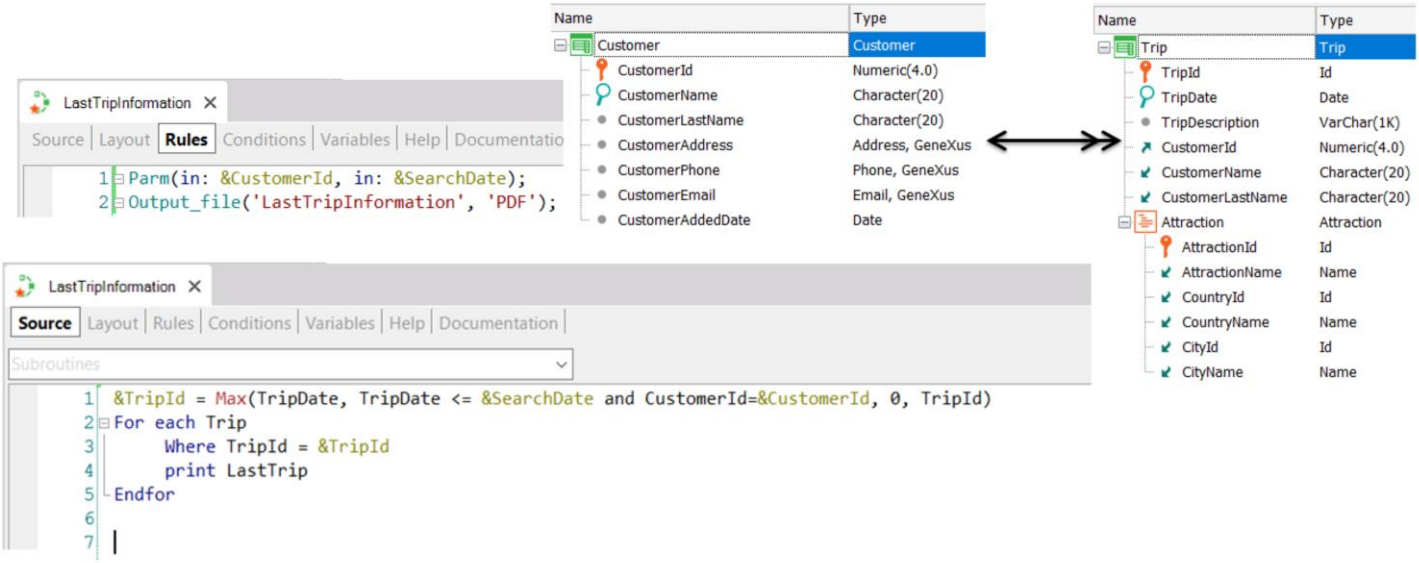
Na fórmula Max, o atributo a maximizar é TripDate, pois queremos a viagem que tenha sido imediatamente anterior a uma data dada, portanto será a viagem com a maior data possível, porém menor ou igual à data de pesquisa.
A condição de agregação especifica que a viagem deve ter uma data menor ou igual à data dada e que também seja do cliente que procura a informação. O valor padrão retornado se não for encontrada nenhuma viagem que atenda a essas condições é 0 e no caso em que se encontre, será o identificador da viagem encontrada.

Agora vamos analisar o que programamos. O For Each irá iterar sobre a tabela TRIP e estabelecerá esse contexto para a fórmula Max.
A fórmula Max também irá iterar na tabela TRIP, mas devido ao contexto, será estabelecido

um filtro automático, uma vez que ambas as tabelas estão relacionadas. Neste caso, é a mesma tabela, então a fórmula ficará filtrada pelo TripId que está posicionado o For Each. Portanto, sempre retornará esse TripId em que o for each se encontra, e é como não ter escrito nada. Ou seja, não nos serve.

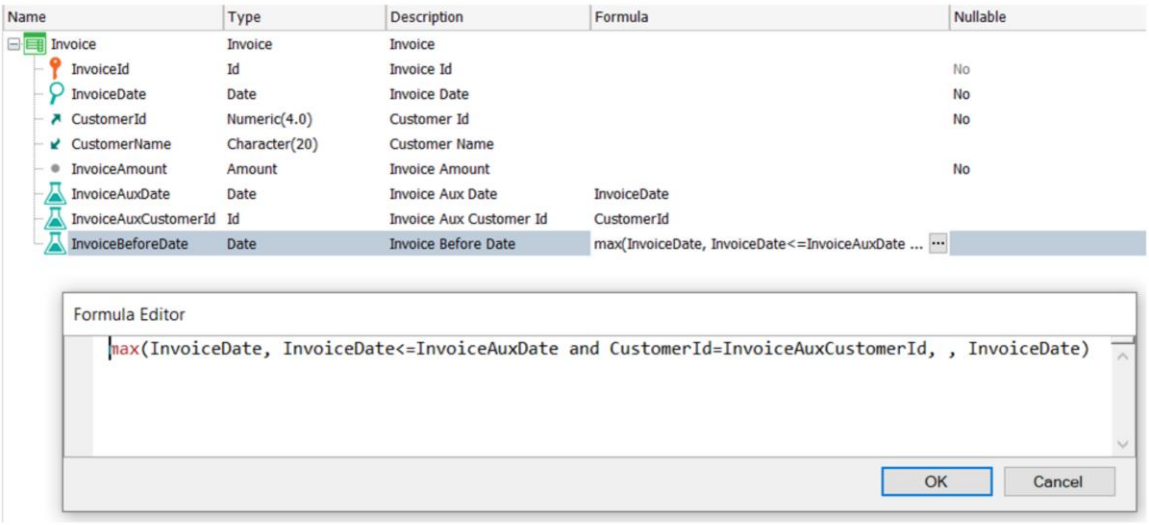
Em suma, devemos ter cuidado quando uma fórmula agreggate navega a mesma tabela que a de seu contexto, pois na maioria das vezes, se não explicitamos o contrário, aplicará um filtro por chave primária, que terá como efeito que a tabela navegada não seja navegada em absoluto e sempre recupere o mesmo registro em que se estava posicionado

Exemplo 2: tabela associada = tabela navegada



A solução neste caso, é executar primeiro a fórmula para encontrar o identificador da viagem que atenda às condições desejadas e, em seguida, filtramos o For Each por esse valor de identificador.

Exemplo 3: tabela associada = tabela navegada

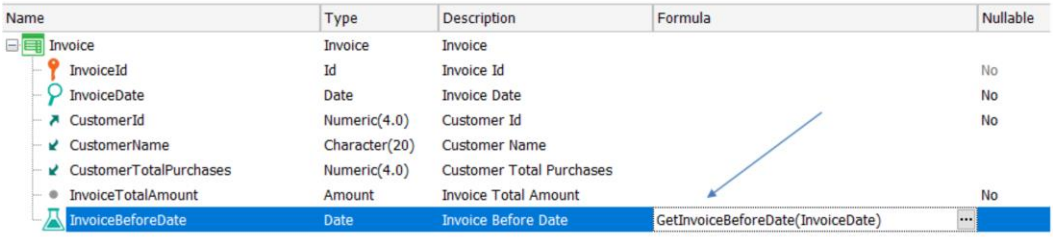


Agora vejamos um caso semelhante ao anterior em uma fórmula global em que dada uma fatura de um cliente, deseja-se encontrar a data da fatura anterior do mesmo cliente. Para resolver isto, utilizaremos uma fórmula Max semelhante à que definimos antes, mas definindo um atributo como fórmula global.

Observemos que novamente estamos tentando definir uma fórmula que navegará sobre a mesma tabela associada à fórmula, então mais uma vez a fórmula ficará filtrada por seu contexto e somente navegará por um único registro!

a diferença entre este caso, que é uma fórmula global, e o anterior que era uma fórmula inline, é que no anterior podíamos disparar a fórmula por fora da tabela de contexto para quebrar esse filtro implícito, mas no caso de uma fórmula global isto não é possível, então NUNCA poderemos ter uma fórmula aggregate global que navegue a mesma tabela.

Exemplo 3: tabela associada = tabela 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)	

As fórmulas de agregação não precisam de um contexto, mas se houver, pode filtrar o resultado

Para isso teremos, de qualquer forma, que defini-la como fórmula horizontal e realizar o cálculo dentro de um procedimento.

Em resumo, verifica-se o que dissemos antes, que embora as fórmulas aggregate não precisem de um contexto para serem avaliadas (como necessitam as horizontais), caso esse contexto exista, não serão considerados todos os registros que a fórmula estabelece explicitamente, mas apenas aqueles que também coincidam com as condições implícitas que surgem desse contexto.

E é importante verificar que o contexto não invalide o objetivo que buscávamos com a fórmula, como vimos nos últimos exemplos apresentados.

GeneXus™

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