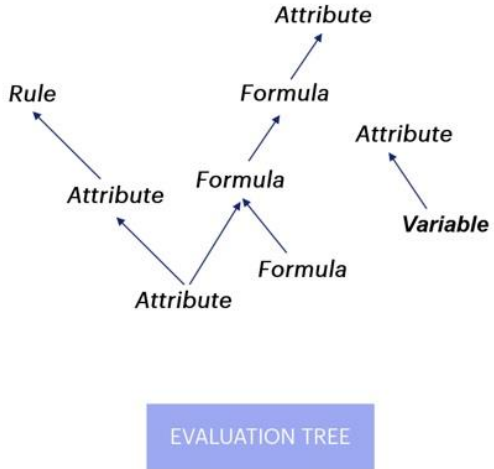


Eventos de disparo de regras em transações

Final

GeneXus

Rule triggering events in Transactions



EVALUATION TREE

on BeforeUpdate
on BeforeInsert *on BeforeDelete*

on AfterUpdate
on AfterInsert *on AfterDelete*

on AfterLevel Level att

on BeforeComplete

on AfterComplete

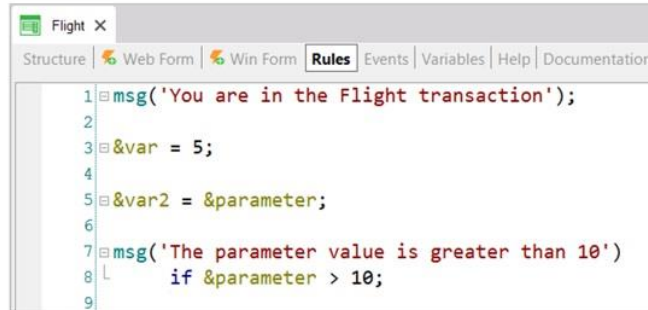
TRIGGERING EVENTS

Ao longo dos vídeos anteriores estudamos a árvore de avaliação que GeneXus cria para estabelecer a ordem de execução de regras e fórmulas, bem como os eventos de disparo que dispomos para condicionar a execução das regras quando a ordem estabelecida pela árvore de avaliação não é o que precisamos.

Neste vídeo revisaremos e reafirmaremos os conhecimentos adquiridos, e veremos um novo evento de disparo que ainda não analisamos.

Stand-alone rules

- THEY ARE TRIGGERED AS SOON AS THE TRN IS EXECUTED
- THERE ARE NO CONDITIONS SET FOR THEIR EXECUTION, AND THEY DON'T HAVE TO WAIT FOR DATA IN ORDER TO BE EXECUTED



```
Flight X
Structure | Web Form | Win Form | Rules | Events | Variables | Help | Documentation
1 msg('You are in the Flight transaction');
2
3 &var = 5;
4
5 &var2 = &parameter;
6
7 msg('The parameter value is greater than 10')
8   if &parameter > 10;
9
```

As primeiras regras a serem disparadas são as chamadas stand-alone, que são aquelas regras que não dependem de nada para serem executadas, ou que já possuem as informações necessárias fornecidas pelos parâmetros recebidos.

Rule triggering events in Transactions

Airport Name Ezeiza

Price

Discount Percentange

Airline Id

Airline Name TAM

Capacity 8

Seat

	Id	Char	Location
✖	1	A	<input type="text" value="Window"/>
✖	2	B	<input type="text" value="Window"/>
✖	3	A	<input type="text" value="Middle"/>
✖	4	B	<input type="text" value="Aisle"/>

- THEY ARE TRIGGERED AS SOON AS THE TRN IS EXECUTED
- THERE ARE NO CONDITIONS SET FOR THEIR EXECUTION, AND THEY DON'T HAVE TO WAIT FOR DATA IN ORDER TO BE EXECUTED

RULES AND FORMULAS TO THE EXTENT THAT ASSOCIATED DATA FROM THE 1ST LEVEL IS AVAILABLE

Após a execução das regras stand-alone, são executadas as regras e fórmulas associadas ao primeiro nível da transação que não estão condicionadas a eventos, de acordo com a árvore de avaliação, na medida em que vá dispondo dos valores envolvidos para executá-las;

Rule triggering events in Transactions

The screenshot shows a transaction form with the following fields:

- Airport Name: Ezeiza
- Price: 1200
- Discount Percentange: 5
- Airline Id: 1
- Airline Name: TAM
- Capacity: 8

The 'Seat' section is highlighted with an orange border and contains a table with the following data:

	Id	Char	Location
✖	1	A	Window
✖	2	B	Window
✖	3	A	Middle
✖	4	B	Aisle

At the bottom of the form are three buttons: CONFIRM, CANCEL, and DELETE.

- THEY ARE TRIGGERED AS SOON AS THE TRN IS EXECUTED
- THERE ARE NO CONDITIONS SET FOR THEIR EXECUTION, AND THEY DON'T HAVE TO WAIT FOR DATA IN ORDER TO BE EXECUTED

RULES AND FORMULAS TO THE EXTENT THAT ASSOCIATED DATA FROM THE 1ST LEVEL IS AVAILABLE

RULES AND FORMULAS TO THE EXTENT THAT ASSOCIATED DATA FROM THE 2ND LEVEL IS AVAILABLE.

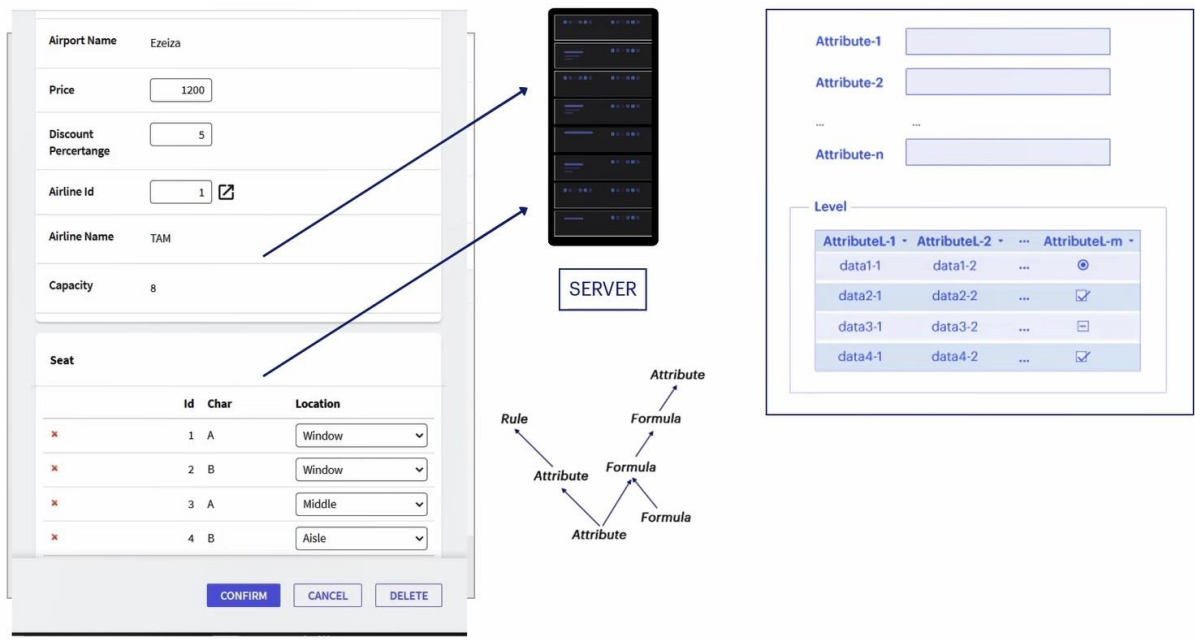
e ao trabalhar no segundo nível, para cada linha, são executadas as regras e fórmulas associadas a esse nível da transação, com o mesmo critério.

Rule triggering events in Transactions



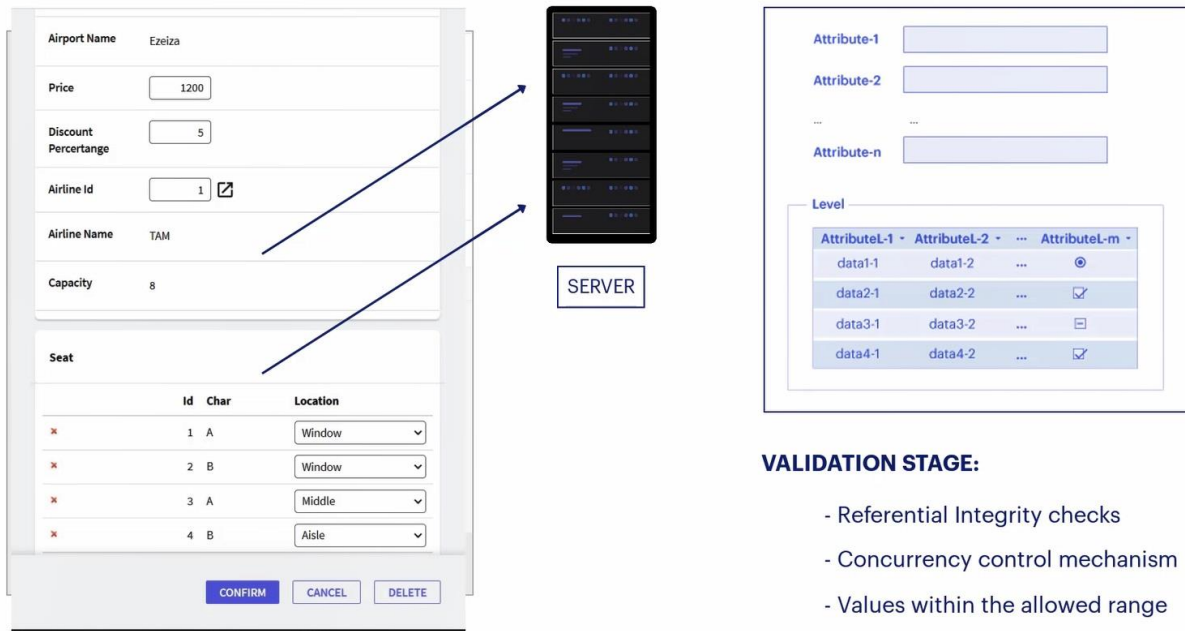
Depois que o usuário pressiona Confirm, os dados trafegam do browser para o servidor web, que volta a percorrer o form como se fosse o usuário, passando um a um em cada campo. Começa a trabalhar com o cabeçalho, disparando aquelas regras que não dependem de mais nada, ou aquelas que dependem do modo (como, por exemplo, a regra Default).

Rule triggering events in Transactions



Em seguida, vai passando por cada campo e vão sendo disparadas as regras correspondentes de acordo com as dependências encontradas.

Rule triggering events in Transactions



Uma vez que GeneXus terminou de executar estas regras e fórmulas, inicia-se uma etapa de validação, na qual são verificados todos os controles de integridade referencial, verifica-se através do mecanismo de controle de concorrência que os dados relevantes não foram modificados e valida-se que os valores que foram inseridos para os atributos estão dentro de seu intervalo permitido.

Rule triggering events in Transactions

Airport Name Ezeiza

Price 1200

Discount Percertange 5

Airline Id 1

Airline Name TAM

Capacity 8

Seat

	Id	Char	Location
✘	1	A	Window
✘	2	B	Window
✘	3	A	Middle
✘	4	B	Aisle

CONFIRM CANCEL DELETE



SERVER

→ **on BeforeValidate Event**

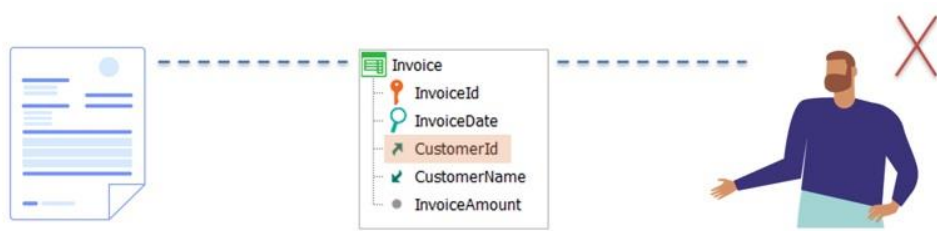
VALIDATION STAGE:

- Referential Integrity checks
- Concurrency control mechanism
- Values within the allowed range

→ **on AfterValidate Event**

GeneXus nos permite condicionar as regras para que sejam executadas imediatamente antes de realizar esta validação, utilizando o evento de disparo on BeforeValidate, e imediatamente depois, utilizando o evento de disparo on AfterValidate.

On BeforeValidate event



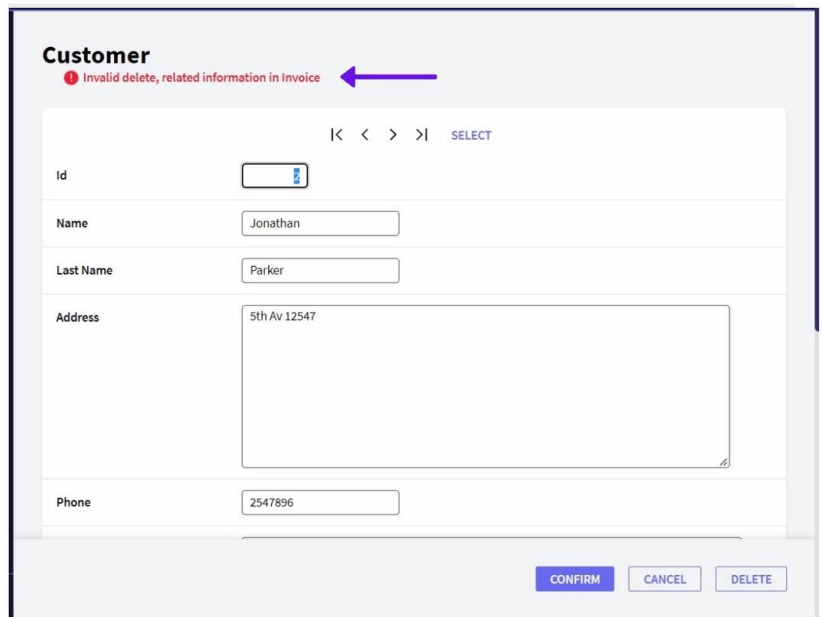
Selection List Invoice			
	Date	Customer Id	Amount
✓	03/28/21	3	3422
✓	01/05/21	1	5200
✓	02/24/21	3	900

Selection List Customer			
	Id	Name	Last Name
✓	1	Joseph	Smith
✓	2	Robert	Jones
✓	3	Anne	Brown

Vejamos um exemplo de uso de BeforeValidate:

Suponhamos que queremos remover um cliente do sistema, que tem uma fatura associada:

On BeforeValidate event



Customer

Invalid delete, related information in Invoice ←

|< > >| SELECT

Id	<input type="text"/>
Name	<input type="text" value="Jonathan"/>
Last Name	<input type="text" value="Parker"/>
Address	<input type="text" value="5th Av 12547"/>
Phone	<input type="text" value="2547896"/>

CONFIRM CANCEL DELETE

Quando tentarmos, veremos esta mensagem de erro, que nos indica que não é possível realizar a exclusão pois existe um registro relacionado na tabela Invoice.

On BeforeValidate event

```

DeleteRelatedInformation
Source  Layout | Rules | Conditions | Variables | Help | Docum
Subroutines
1 For each Invoice
2   where CustomerId = &CustomerId
3   delete
4 Endfor

```

```

DeleteRelatedInformation  Customer
Structure  Web Layout | Win Form | Rules | Events | Vari
1 DeleteRelatedInformation(CustomerId)
2   if delete
3     on BeforeValidate;

```

Customer

Invalid delete, related information in invoice ←

|< < > >| SELECT

Id

Name

Last Name

Address

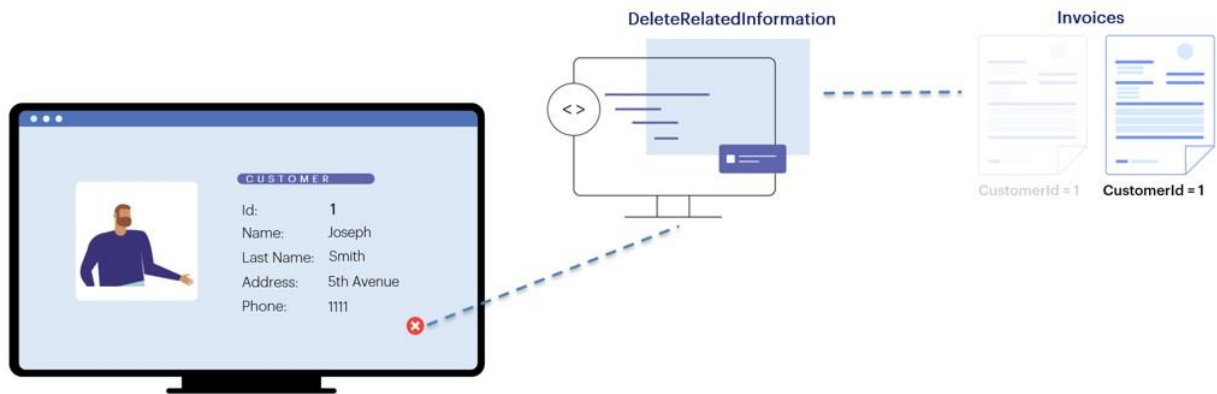
Phone

CONFIRM CANCEL DELETE

Para resolver isso, podemos chamar um procedimento que remova as informações relacionadas da tabela correspondente, de forma que não haja violações na integridade dos dados ao tentar efetuar a eliminação.

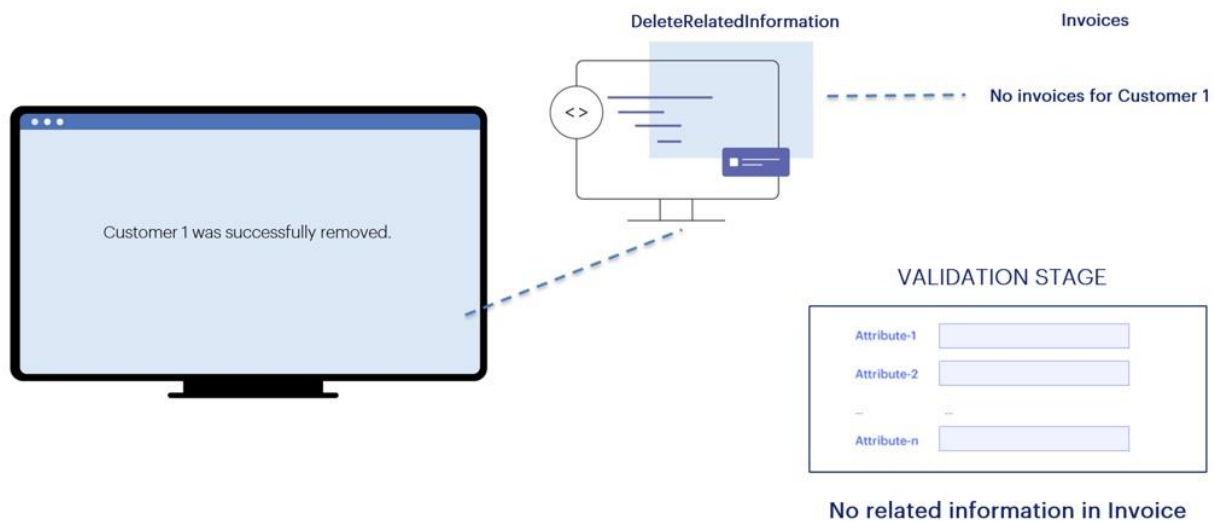
Este procedimento deve ser executado antes que GeneXus comece a realizar as verificações de integridade referencial que, como acabamos de mencionar, é durante a etapa de validação, portanto condicionaremos a regra para que seja executada no evento on BeforeValidate:

On BeforeValidate event



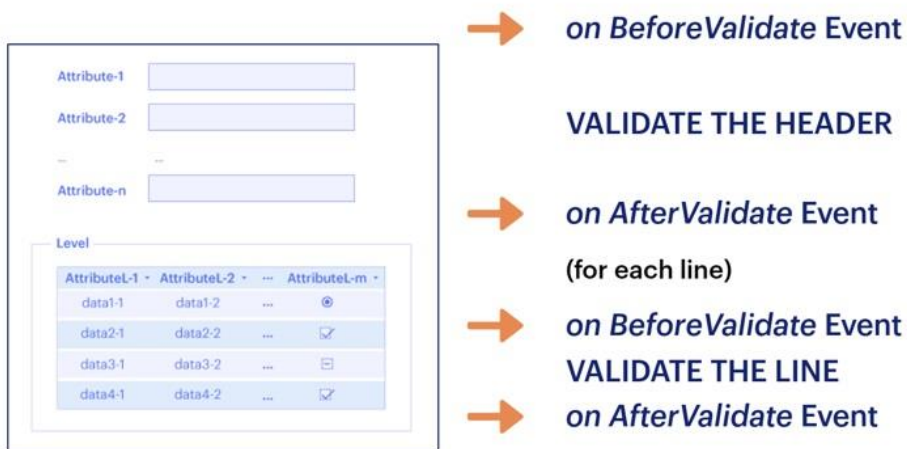
Desta forma, quando vamos remover o cliente com id 1, antes de iniciar as verificações de validação, será invocado o procedimento, que elimina as possíveis faturas associadas a esse cliente,

On BeforeValidate event



portanto, quando realiza a validação, já não encontra registros relacionados e é efetuada a remoção do cliente.

On BeforeValidate / On AfterValidate



Resumindo, o evento de disparo BeforeValidate ocorre um instante de tempo antes que a informação da instância com a qual se está trabalhando (cabeçalho ou linha) seja validada. Ou seja, ocorrerá um instante de tempo antes da ação de “validação do cabeçalho” ou “validação da linha”, conforme o caso. Lembre-se que aqui também já terão sido disparadas todas as regras que não estão condicionadas a nenhum evento de disparo e que estão relacionadas ao nível; e o evento de disparo on AfterValidate permite especificar que uma regra seja executada imediatamente após terem sido validados os dados dessa instância, antes de serem gravados fisicamente na tabela correspondente.

Todas as regras que estão condicionadas ao evento on BeforeValidate ou on AfterValidate serão disparadas nestes momentos precisos. Em que ordem? Na qual estão escritas, a menos que haja dependências entre elas, como já vimos.

Rule triggering events in Transactions

Attribute-1 ✓

Attribute-2 ✓

...

Attribute-n ✓

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	⊕
data2-1	data2-2	...	✓
data3-1	data3-2	...	⊖
data4-1	data4-2	...	✓

on BeforeValidate

VALIDATE THE HEADER

on AfterValidate

on BeforeInsert / on BeforeUpdate / on BeforeDelete

Em seguida, se a validação não falhou e:

- a gravação correspondia a uma inserção: serão executadas as regras associadas ao primeiro nível da transação com evento de disparo BeforeInsert.
- se a gravação correspondia a uma atualização: serão executadas as regras associadas ao primeiro nível da transação com evento de disparo BeforeUpdate.
- e se a gravação correspondia a uma exclusão: serão executadas as regras associadas ao primeiro nível da transação com evento de disparo BeforeDelete.

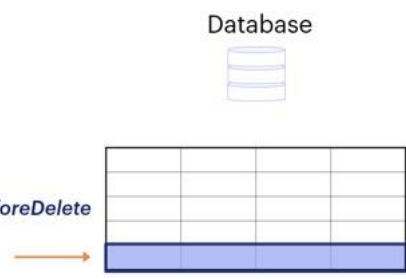
Rule triggering events in Transactions

Attribute-1 ✓
Attribute-2 ✓
...
Attribute-n ✓

Level

AttributeL-1	AttributeL-2	...	AttributeL-m
data1-1	data1-2	...	⊕
data2-1	data2-2	...	✓
data3-1	data3-2	...	☐
data4-1	data4-2	...	✓

on BeforeValidate
VALIDATE THE HEADER
on AfterValidate
on BeforeInsert / on BeforeUpdate / on BeforeDelete
SAVE THE HEADER



Em seguida, será efetuada a gravação dos dados do cabeçalho na tabela correspondente...

Rule triggering events in Transactions



- e a Gravação da linha, portanto serão disparadas todas as regras que estão condicionadas aos eventos BeforeInsert (ou BeforeUpdate, ou BeforeDelete) e AfterInsert (ou AfterUpdate, ou AfterDelete) que envolvem atributos do segundo nível

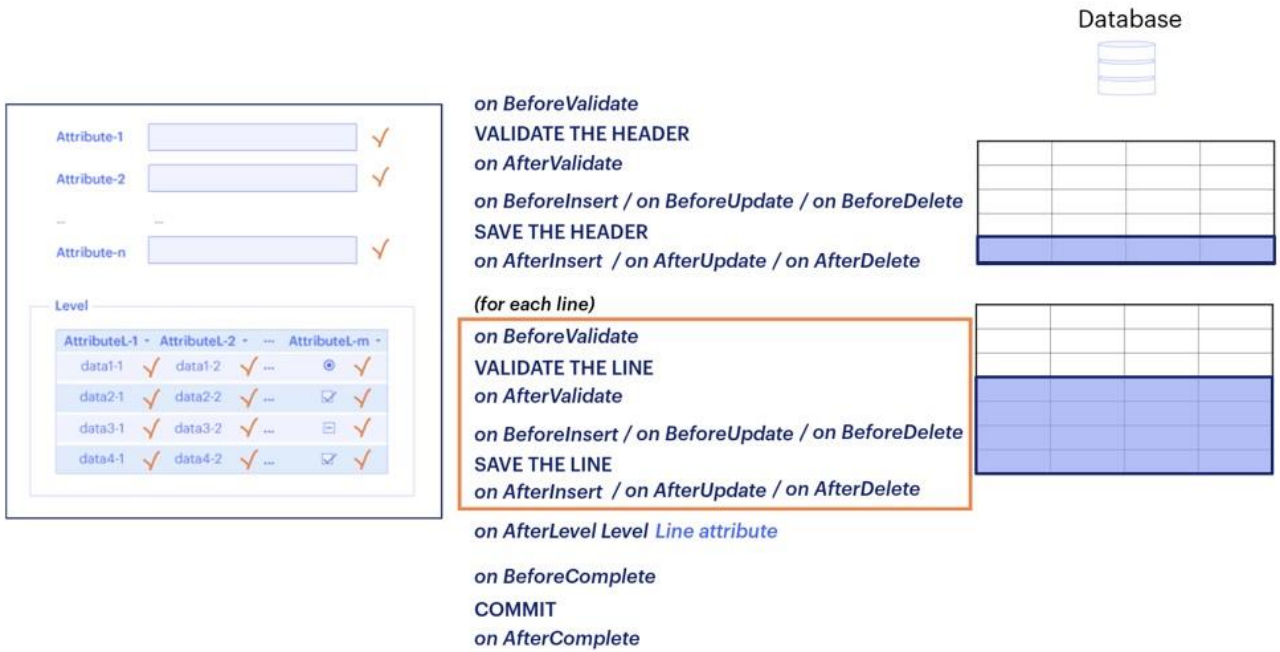
Rule triggering events in Transactions



Então, serão executadas as regras com evento de disparo “on AfterLevel” para fazer ou avaliar algo imediatamente após percorrer um determinado nível;

Se houvesse outro grid, ou seja, outro nível, seria repetido o mesmo que foi feito para o primeiro grid e depois também ocorrerá o on AfterLevel, mas deste outro nível. E assim sucessivamente até terminar com o último grid.

Rule triggering events in Transactions



Após terem sido executadas todas as operações explicadas até o momento, será realizado um COMMIT, que se encarregará de consolidar na base de dados, os dados do cabeçalho e todas as linhas da transação. Contamos com o evento "on BeforeComplete" para executar regras imediatamente antes de ser efetuado o Commit, e "on AfterComplete", que corresponde ao instante de tempo imediatamente posterior.

Rule triggering events in Transactions



É importante assimilar a ordem em que são executadas as regras em uma transação, quais são os eventos de disparo disponíveis para lhes atribuir, quando se disparam exatamente, e quais ações ocorrem antes e depois de cada evento de disparo, já que apenas conhecendo-os bem será possível programar o comportamento das transações adequadamente.

*GeneXus*TM

training.genexus.com
wiki.genexus.com