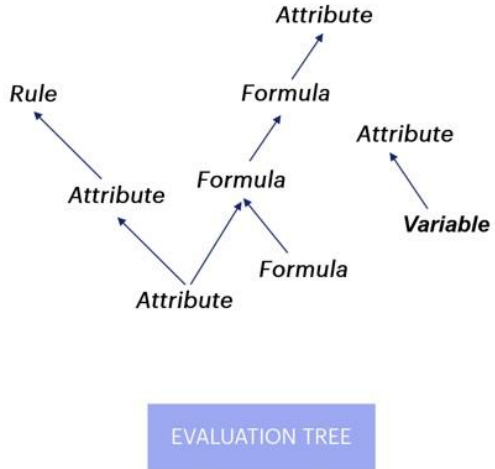


Eventos de disparo de reglas en transacciones

Final

GeneXus[™]

Rule triggering events in Transactions

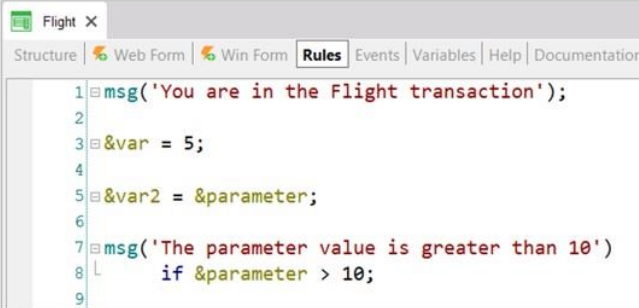


A lo largo de los videos anteriores hemos estudiado el árbol de evaluación que crea GeneXus para establecer el orden de ejecución de reglas y fórmulas, así como también los eventos de disparo de los que disponemos para condicionar la ejecución de las reglas cuando el orden establecido por el árbol de evaluación no es el que necesitamos.

En este video repasaremos y reafirmaremos los conocimientos adquiridos, y veremos un nuevo evento de disparo que aún no hemos analizado.

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



```
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
```

Las primeras reglas en dispararse son las llamadas stand-alone, que son aquellas reglas que no dependen de nada para ejecutarse, o que ya cuentan con la información necesaria provista por los parámetros recibidos.

Rule triggering events in Transactions

Airport Name Ezeiza

Price 1200

Discount Percentange 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

- 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

Luego de la ejecución de las reglas stand-alone, se ejecutan las reglas y fórmulas asociadas al primer nivel de la transacción que no estén condicionadas a eventos, de acuerdo al árbol de evaluación, en la medida de que se vaya disponiendo de los valores involucrados para ejecutarlas;

Rule triggering events in Transactions

Airport Name	Ezeiza
Price	<input type="text" value="1200"/>
Discount Percentange	<input type="text" value="5"/>
Airline Id	<input type="text" value="1"/>
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

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

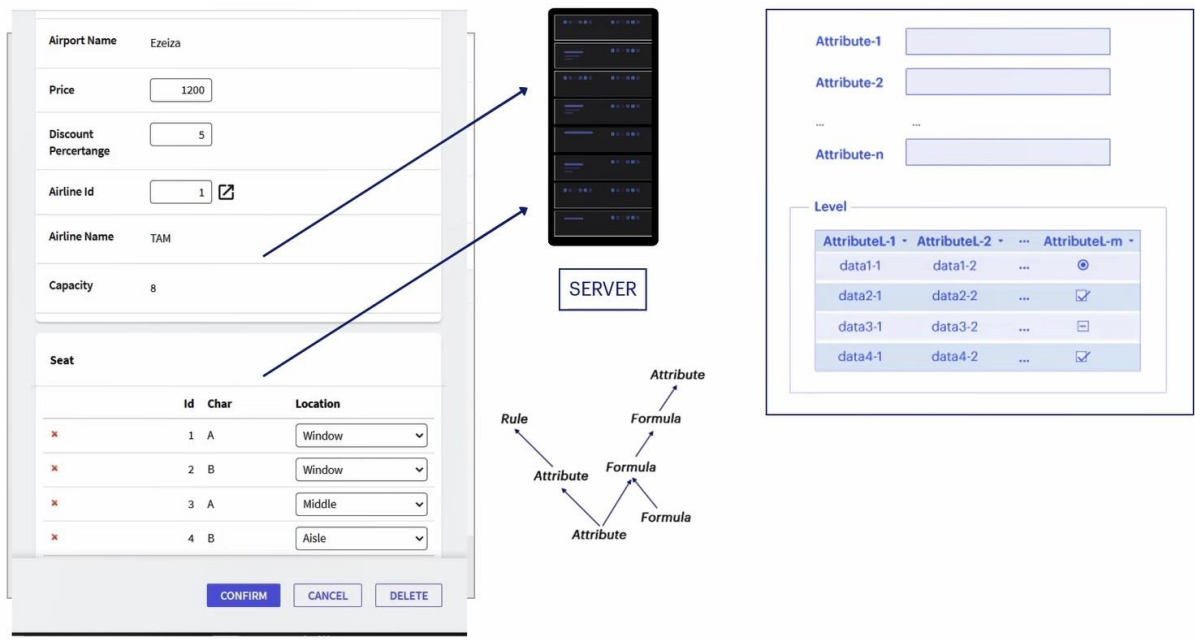
y al trabajar en el segundo nivel, para cada línea, se ejecutan las reglas y fórmulas asociadas a ese nivel de la transacción, con el mismo criterio.

Rule triggering events in Transactions



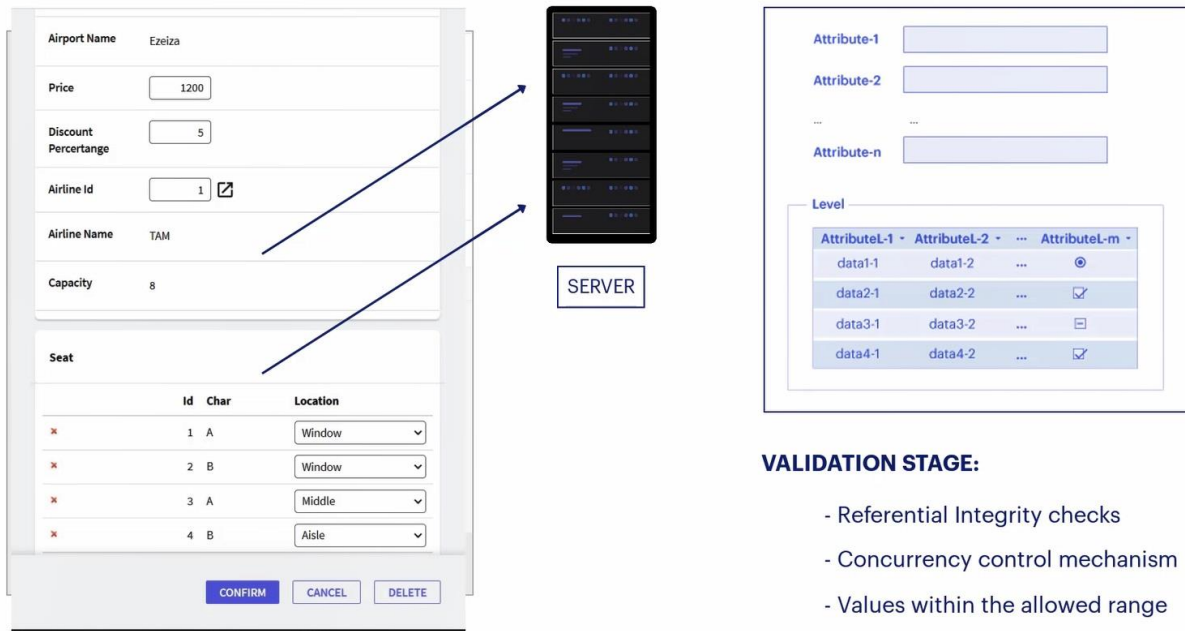
Después de que el usuario presiona Confirm, los datos viajan desde el browser al servidor web, que vuelve a recorrer el form como si se tratara del usuario pasando uno a uno por cada campo. Comienza a trabajar con el cabezal, disparando aquellas reglas que no dependen de nada más, o las que dependen del modo (como por ejemplo, la regla Default).

Rule triggering events in Transactions



Luego se va pasando por cada campo y se van disparando las reglas correspondientes de acuerdo a las dependencias encontradas.

Rule triggering events in Transactions



VALIDATION STAGE:

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

Una vez que GeneXus terminó de ejecutar estas reglas y fórmulas, comienza una etapa de validación, en la que se chequean todos los controles de integridad referencial, se verifica mediante el mecanismo de control de concurrencia que los datos relevantes no hayan sido modificados, y se valida que los valores que fueron introducidos para los atributos estén dentro de su rango 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**

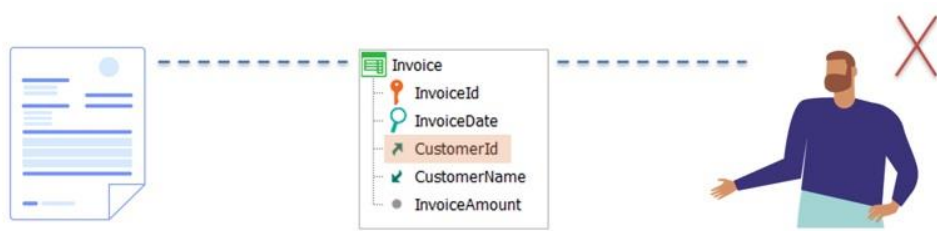
VALIDATION STAGE:

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

→ **on AfterValidate Event**

GeneXus nos permite condicionar las reglas para que sean ejecutadas inmediatamente antes de realizar esta validación, utilizando el evento de disparo on BeforeValidate, e inmediatamente después, utilizando el evento de disparo on AfterValidate.

On BeforeValidate event



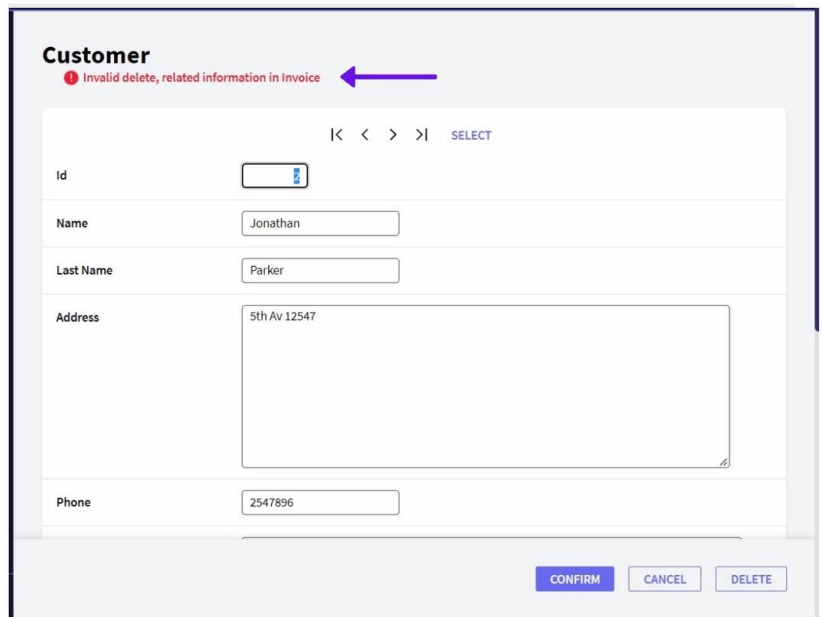
Selection List Invoice			
	Date	Customer Id	Amount
✓	03/28/21	3	3422
✓	01/05/21	1	5200
✓	02/24/21	3	900

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

Veamos un ejemplo de uso de BeforeValidate:

Supongamos que queremos eliminar un cliente del sistema, que tiene asociada una factura:

On BeforeValidate event



The screenshot shows a web form titled "Customer". At the top left, there is a red error message: "Invalid delete, related information in Invoice". A purple arrow points from this message to the "DELETE" button at the bottom right of the form. The form contains several input fields: "Id" (empty), "Name" (Jonathan), "Last Name" (Parker), "Address" (5th Av 12547), and "Phone" (2547896). At the top right of the form area, there are navigation controls: "|< < > >|" and a "SELECT" button. At the bottom right, there are three buttons: "CONFIRM" (highlighted in blue), "CANCEL", and "DELETE".

Al momento de intentarlo, veremos este mensaje de error, que nos indica que no es posible realizar la eliminación ya que existe un registro relacionado en la tabla 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

Navigation: |< < > >| SELECT

Id:

Name:

Last Name:

Address:

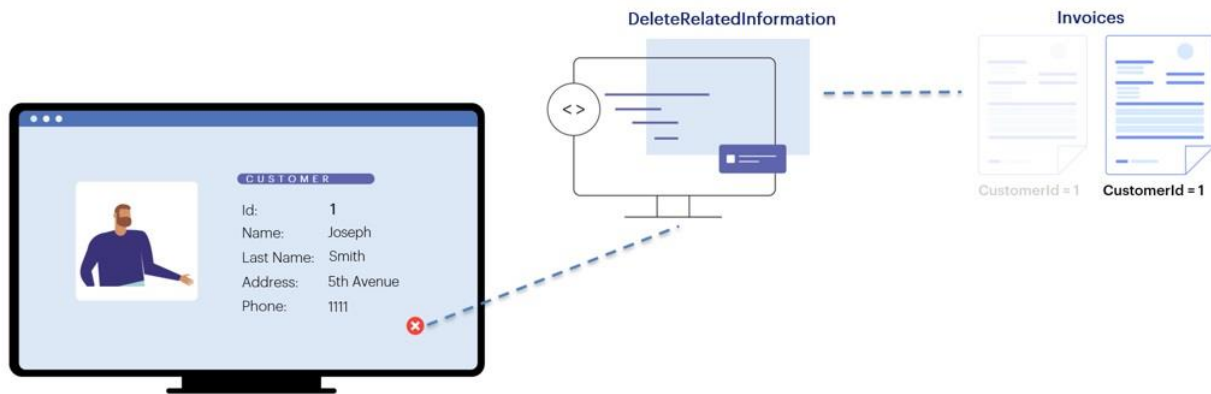
Phone:

Buttons: CONFIRM CANCEL DELETE

Para solucionarlo, podemos llamar a un procedimiento que elimine la información relacionada de la tabla correspondiente, de modo que no haya violaciones en la integridad de los datos a la hora de intentar efectuar la eliminación.

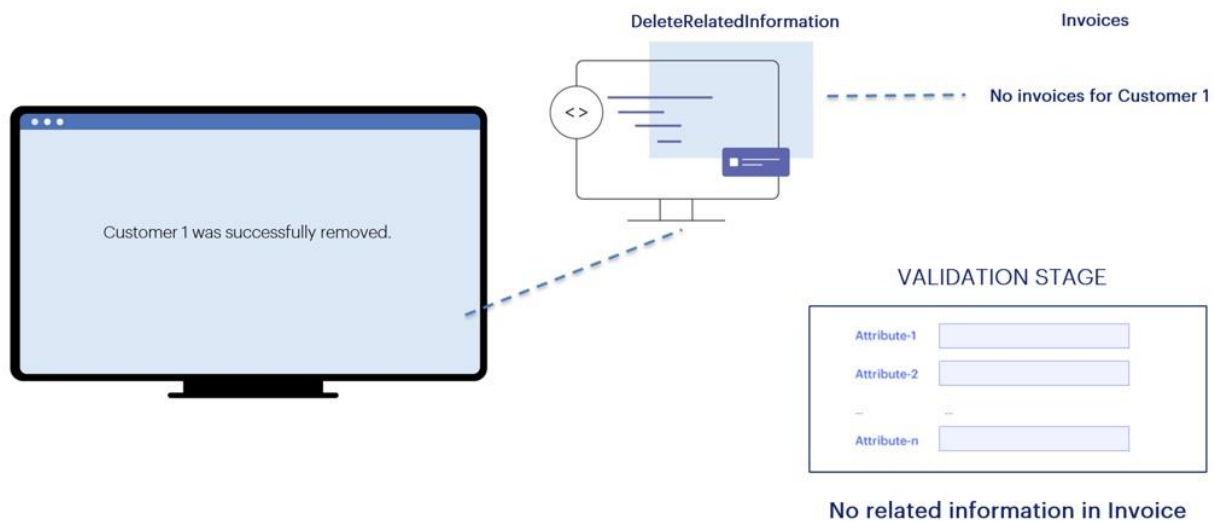
Este procedimiento debe ejecutarse antes de que GeneXus comience a realizar los chequeos de integridad referencial, que, como acabamos de mencionar, es durante la etapa de validación, por lo que condicionaremos la regla para que se ejecute en el evento on BeforeValidate:

On BeforeValidate event



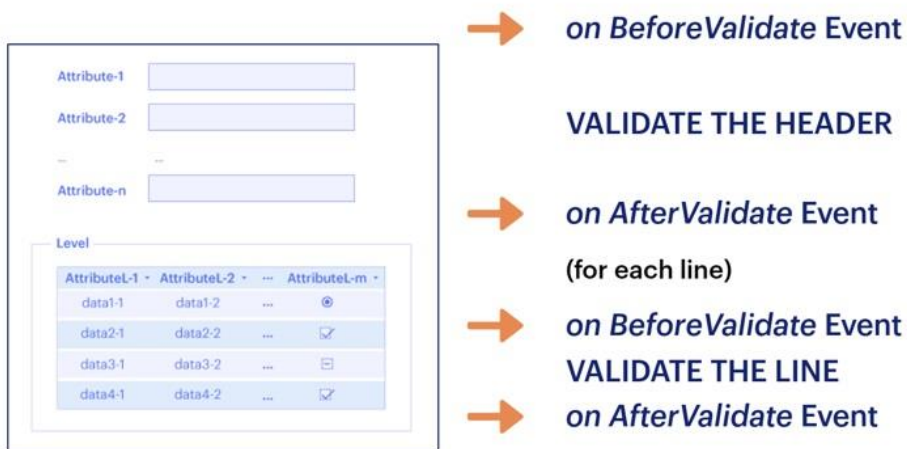
De este modo, cuando vayamos a eliminar al cliente con id 1, antes de comenzar los chequeos de validación, será invocado el procedimiento, que elimina las posibles facturas asociadas a ese cliente,

On BeforeValidate event



por lo que cuando realiza la validación ya no encuentra registros relacionados, y se efectúa la eliminación del cliente.

On BeforeValidate / On AfterValidate



Resumiendo, el evento de disparo BeforeValidate ocurre un instante de tiempo antes de que la información de la instancia con la que se está trabajando (cabezal o línea) sea validada. Es decir, ocurrirá un instante de tiempo antes de la acción de “validación del **cabezal**” o “validación de la **línea**”, según corresponda. Recuerde que aquí también ya se habrán disparado todas las reglas que no estén condicionadas a evento de disparo alguno y que estén relacionadas al nivel; y el evento de disparo on AfterValidate permite especificar que una regla se ejecute inmediatamente después de que se hayan validado los datos de esa instancia, antes de que se grabe físicamente en la tabla correspondiente.

Todas las reglas que estén condicionadas al evento on BeforeValidate u on AfterValidate se van a disparar en estos momentos precisos. ¿En qué orden? En el que estén escritas, a menos que haya dependencias entre ellas, como ya 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

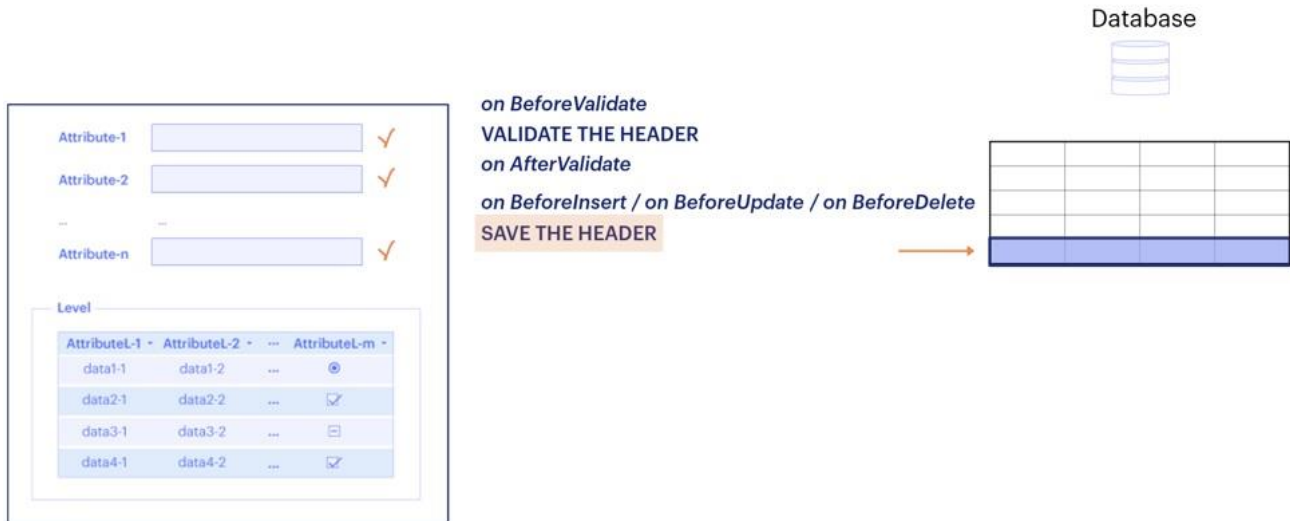
on AfterValidate

on BeforeInsert / on BeforeUpdate / on BeforeDelete

Seguidamente, si la validación no falló y:

- la grabación correspondió a una inserción: se ejecutarán las reglas asociadas al primer nivel de la transacción con evento de disparo BeforeInsert.
- si la grabación correspondió a una actualización: se ejecutarán las reglas asociadas al primer nivel de la transacción con evento de disparo BeforeUpdate.
- y si la grabación correspondió a una eliminación: se ejecutarán las reglas asociadas al primer nivel de la transacción con evento de disparo BeforeDelete.

Rule triggering events in Transactions



Luego se efectuará la grabación de los datos del cabezal en la tabla correspondiente...

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***on AfterInsert / on AfterUpdate / on AfterDelete*

Database



...y a continuación se ejecutarán las reglas asociadas a este primer nivel con evento “on AfterInsert” (si la grabación correspondió a una inserción), “on AfterUpdate” (si la grabación correspondió a una actualización), y “on AfterDelete” (si la grabación correspondió a una eliminación).

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
on AfterInsert / on AfterUpdate / on AfterDelete

(for each line)
on BeforeValidate
VALIDATE THE LINE
on AfterValidate

Database



Si la transacción es de dos niveles, entonces luego de la grabación del cabezal, se ejecutarán las reglas y fórmulas de atributos del segundo nivel que no tengan evento de disparo asociado, y a continuación se producirá para cada línea:

- la Validación, por lo que se dispararán todas las reglas que estén condicionadas a los eventos BeforeValidate y AfterValidate que involucren atributos del segundo nivel

Rule triggering events in Transactions



- y la Grabación de la línea, por lo que se dispararán todas las reglas que estén condicionadas a los eventos BeforeInsert (o BeforeUpdate, o BeforeDelete) y AfterInsert (o AfterUpdate, o AfterDelete) que involucren atributos del segundo nivel

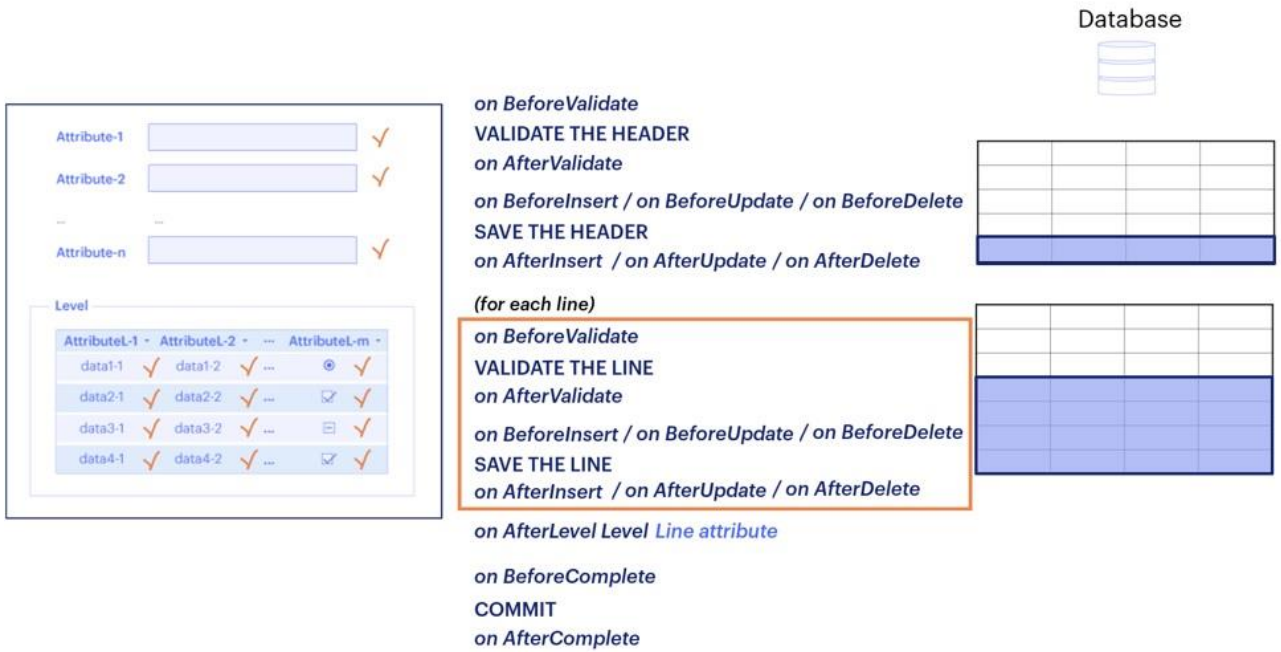
Rule triggering events in Transactions



Luego se ejecutarán las reglas con evento de disparo “on AfterLevel” para hacer o evaluar algo inmediatamente después de recorrer un determinado nivel;

Si hubiera otro grid, es decir, otro nivel, se repetiría lo mismo que se hizo para el primer grid y luego también ocurrirá el on AfterLevel pero de este otro nivel. Y así sucesivamente hasta terminar con el último grid.

Rule triggering events in Transactions



Luego de haberse ejecutado todas las operaciones explicadas hasta el momento, se efectuará un COMMIT, que se encargará de consolidar en la base de datos los datos del cabezal y todas las líneas de la transacción. Contamos con el evento “on BeforeComplete” para ejecutar reglas inmediatamente antes de efectuado el Commit, y “on AfterComplete”, que corresponde al instante de tiempo inmediatamente posterior.

Rule triggering events in Transactions



Es importante asimilar el orden en el que se ejecutan las reglas en una transacción, cuáles son los eventos de disparo disponibles para asignarles, cuándo se disparan exactamente, y qué acciones ocurren antes y después de cada evento de disparo, ya que solamente conociéndolos bien se podrá programar el comportamiento de las transacciones adecuadamente.

*GeneXus*TM

training.genexus.com
wiki.genexus.com