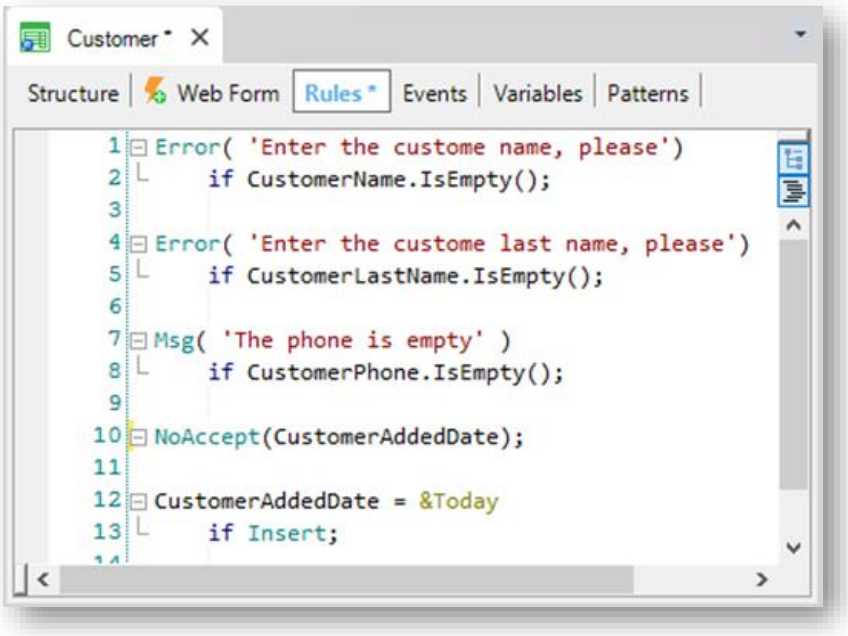
Eventos de disparo de regras em transações

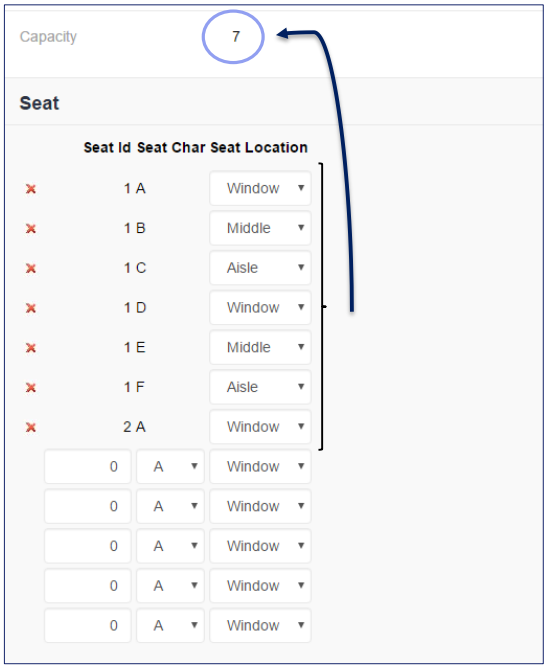
GeneXus 16

Regras



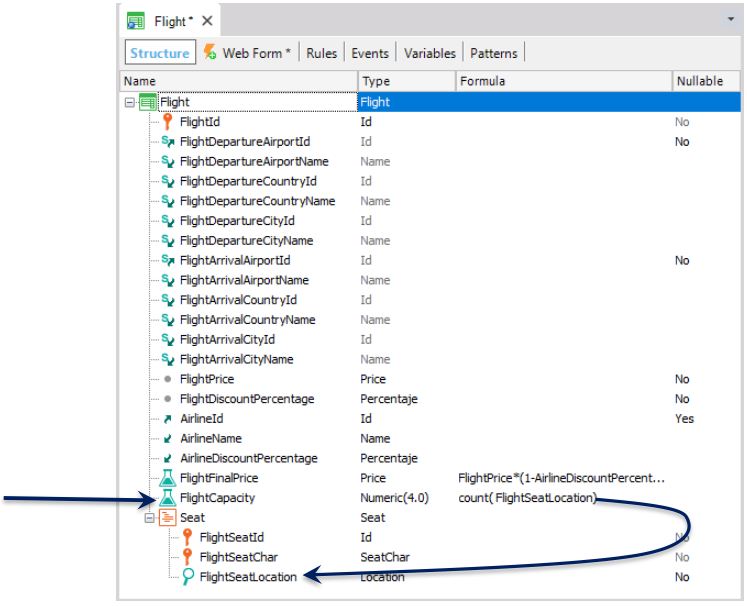
Quando estudamos as regras que podemos escrever nas transações, falamos que não era necessário especificar a ordem de execução de cada uma delas, já que GeneXus determina esses momentos de disparo.

Na maioria das vezes, as regras que definimos são executadas no momento que pretendemos, no entanto, em alguns casos pode ser necessário alterar esse momento.



Vejamos um exemplo. Imagine que para cada voo, a companhia aérea deseje controlar que não possa ser cadastrado uma quantidade de lugares incorreta, por exemplo, que cada voo não possa ter menos de oito lugares. Lembremos que tínhamos o atributo FlightCapacity, fórmula que contava a quantidade de lugares.

Exemplo: número de lugares em um voo

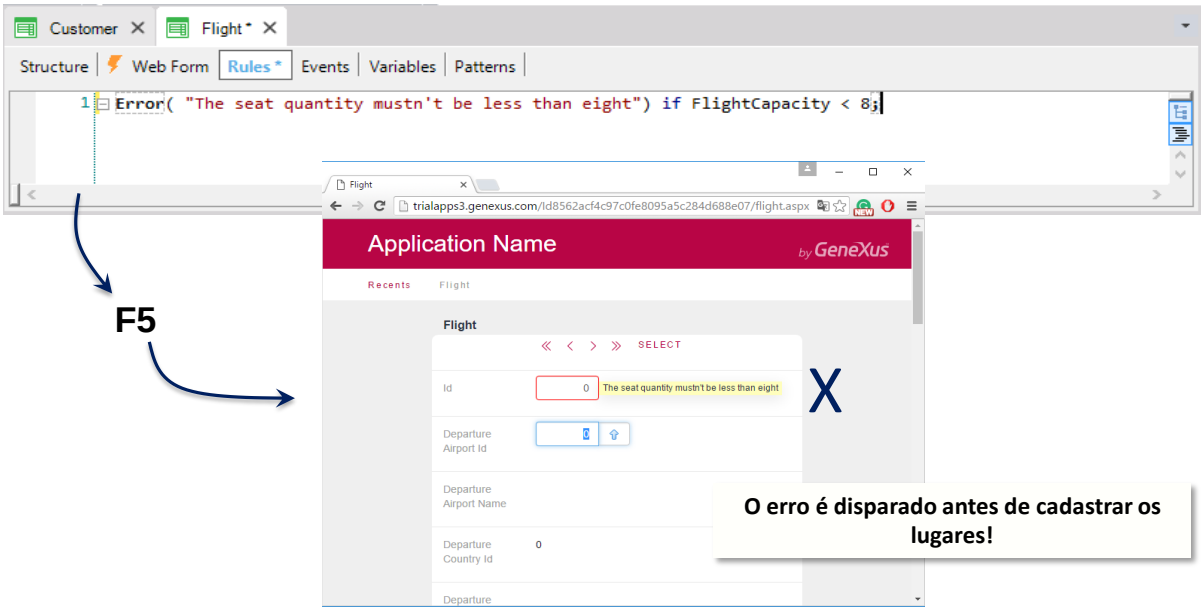


FlightCapacity >= 8

Ao ser cadastrado um novo voo, queremos que seja realizado o controle correspondente e que não seja possível salvar o voo caso não se cumpra a condição desejada.

Para conseguir isso, na transação que cadastra voos, vamos declarar um regra a partir do atributo FlightCapacity que conta a totalidade de lugares do avião.

Exemplo: número de lugares em um voo



Assim sendo, vamos na seção de Rules e declaramos uma regra Error que não permita armazenar um voo com menos de 8 lugares. Escrevemos…

Error… a quantidade de lugares não pode ser menor a oito if FlightCapacity é menor que oito… Terminamos a instrução com ponto-e-vírgula…

Pressionamos F5…

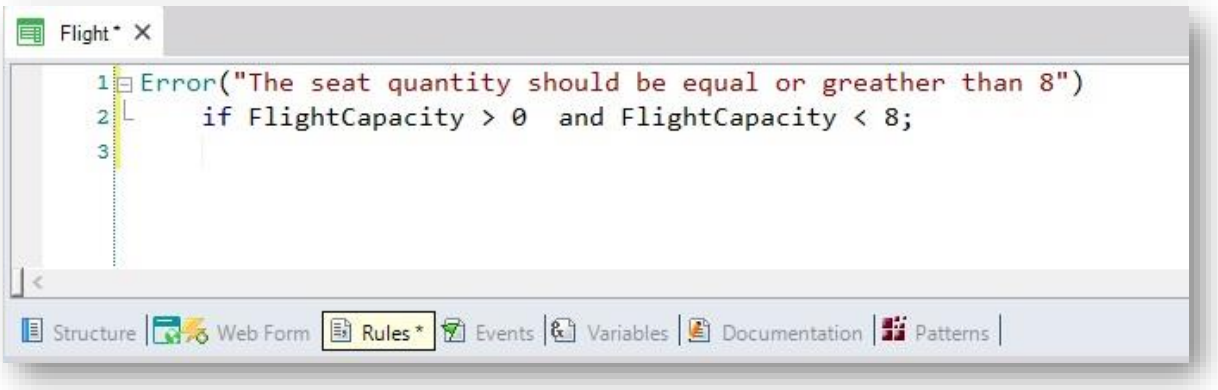
Vamos abrir a transação Flight para criar um novo voo.

Visualizamos que já está sendo disparado o erro.

Por que? Porque a fórmula é disparada logo no carregamento da transação, e vai mudando conforme inserimos linhas. O problema é que não tivemos tempos de cadastrar nenhuma linha, portanto, a fórmula FlightCapacity é disparada dando por resultado zero, que é menor que oito.

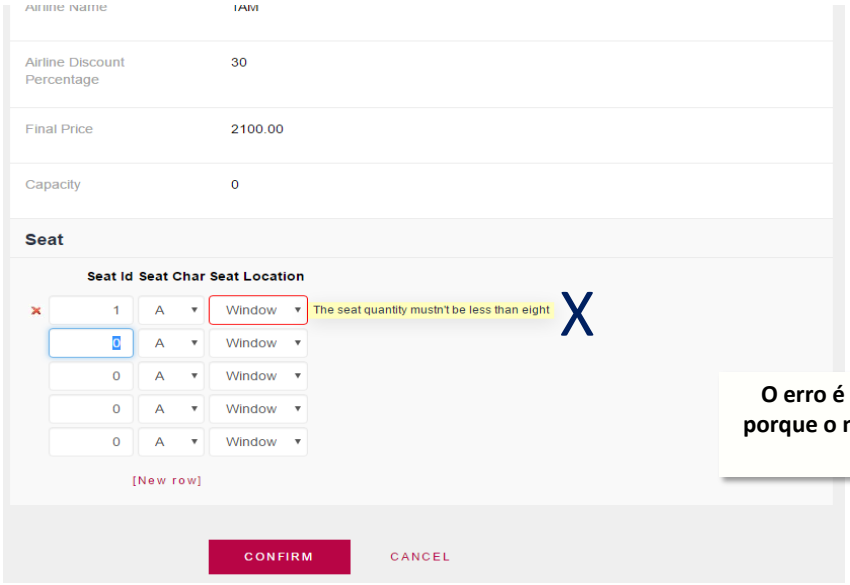
Exemplo: número de lugares em um voo

Poderíamos pensar em modificar a condição de erro:



Podemos, então, condicionar o erro.

Exemplo: número de lugares em um voo



Para dar tempo de cadastrar alguma linha.

Pressionamos F5.

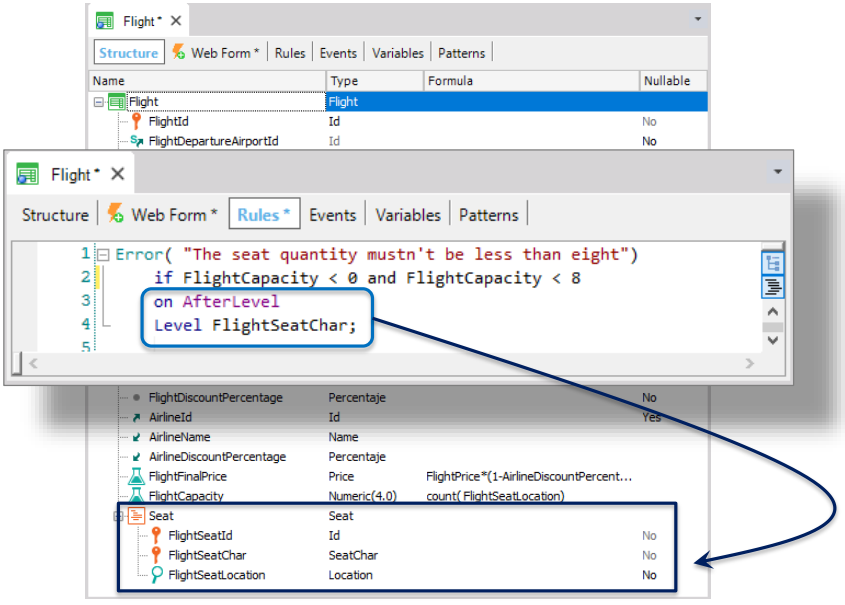
Deixamos o valor do identificador sem preenchimento já que é autonumber…

Cadastramos um voo do aeroporto de Guarulhos, em São Paulo, Brasil…, até o aeroporto Charles de Gaulle em Paris, França. O preço do voo é de 3000, o desconto é de 10% e a companhia aérea é a TAM.

Agora cadastramos o lugar 1, letra A, janela…30 e ao sair da linha, visualizamos a mensagem de erro:

Obviamente não queremos que a mensagem de erro seja disparada aqui, porque todavia não poderemos cadastrar todos os lugares. É claro que se cadastrarmos um único lugar, temos menos de 8 lugares cadastrados, o que corresponde com o disparo da regra Error, mas na realidade necessitamos que o controle da quantidade de lugares seja realizado **depois** que o usuário termine de cadastrar todos os lugares.

Exemplo: número de lugares em um voo



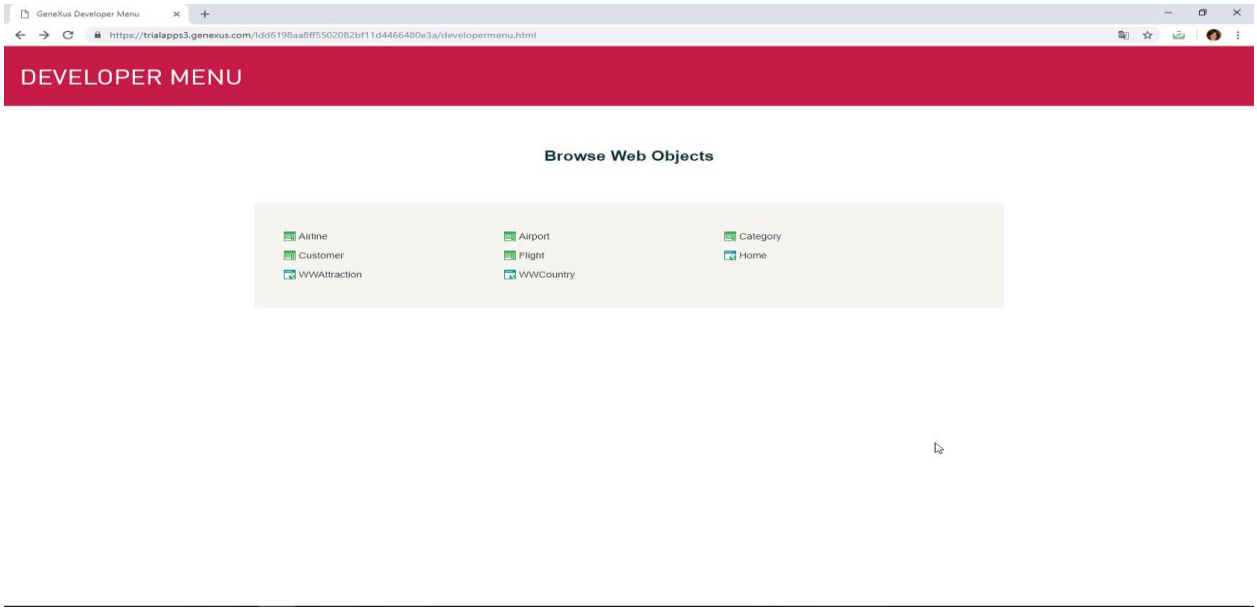
Para conseguir isso, devemos condicionar a regra a disparar-se depois que terminamos de trabalhar com as linhas do grid, para isso escrevemos... On afterlevel level FlightSeatChar.

O momento “on after level” faz com que a regra seja disparada depois de terminar um nível. Em nosso caso será depois de teminar o nível das linhas do grid de lugares, então, incluímos “level FlightSeatChar” já que este atributo está no nível das linhas de lugares. Poderíamos ter utilizado qualquer um dos outros atributos do nível, por exemplo, FlightSeatLocation.

Dessa forma, indicamos para GeneXus que essa regra deve ser disparada **depois** de terminado de cadastrar os dados onde está o atributo FlightSeatChar, ou seja, depois de cadastrar os dados de todos os lugares do voo.

A avaliação feita pela regra Error tem sentido, já que no momento de ser disparada já teremos cadastrados todos os lugares que o usuário deseja e poderá ser verificado que exista pelo menos 8 lugares cadastrados.

DEMO



[DEMO: https://youtu.be/PblNZf47qXM]

Pressionamos F5…

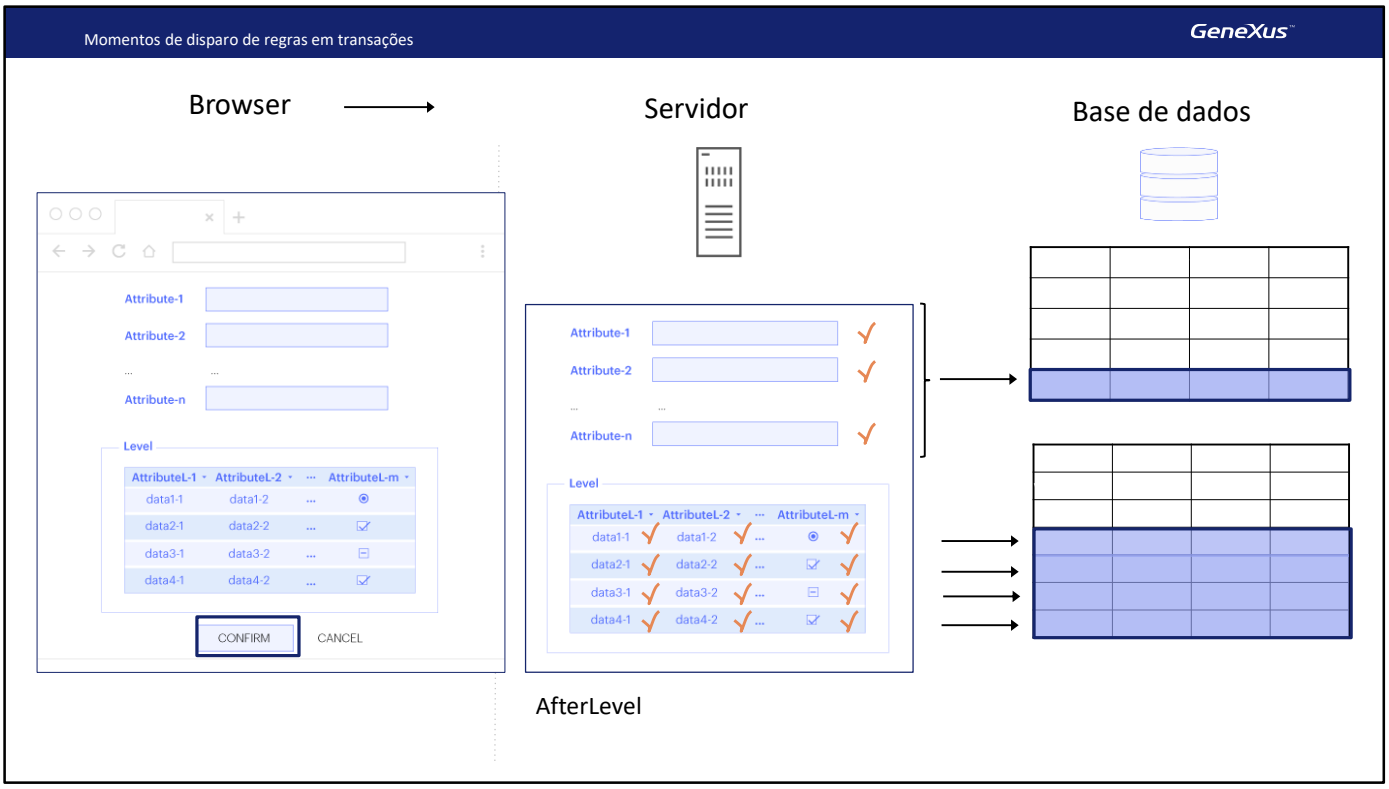
Abrimos a transação Flight e vamos cadastrar um novo voo outra vez.

Repetimos os dados que usamos antes... O voo do aeroporto de Guarulhos, até o aeroporto Charles de Gaulle. O preço de 3000, com 10% de desconto e a companhia aérea TAM.

Agora cadastramos os lugares...

1, A, janela
1, B, corredor
2, A, janela
2, B, corredor

E pressionamos Confirmar, para indicar que terminamos de cadastrar os dados do voo (incluindo os lugares) e que o voo pode ser gravado na base de dados.



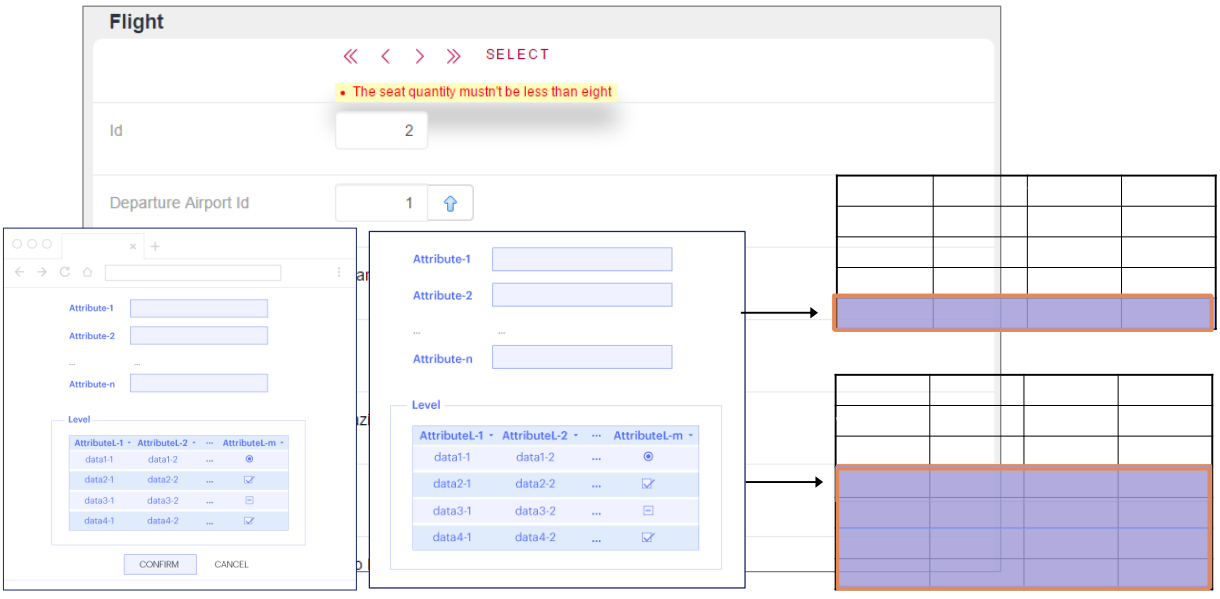
O que ocorre a parti daí? Será enviado os dados do cabeçalho e linhas ao servidor, e serão processados um por um, disparando as regras correspondentes.

Quando terminado a validação dos dados do cabeçalho, será inserido o registro na tabela.

E logo será feito o mesmo para cada linha.

Quando terminado de processar todas as linhas, esse é o momento **AfterLevel**. Aí serão disparadas todas as regras que foram condicionadas para esse momento.

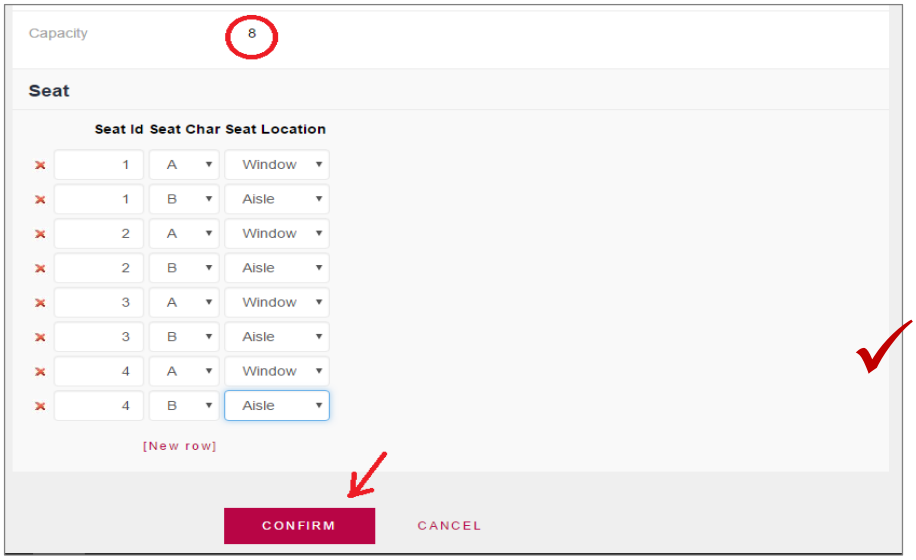
Observemos que já estão gravados os dados do cabeçalho e linhas na base de dados.



Se só cadastramos apenas 4 linhas ... AfterLevel → ERROR

Voltando a nossa transação em execução, ao Confirmar a transação, GeneXus indica-nos o erro, tal como esperávamos, pois cadastramos só quatro lugares e **não** vai gravar este voo na base de dados. É que uma regra de Error desfaz toda a gravação feita anteriormente.

Exemplo: número de lugares em um voo



8 lugares gravados

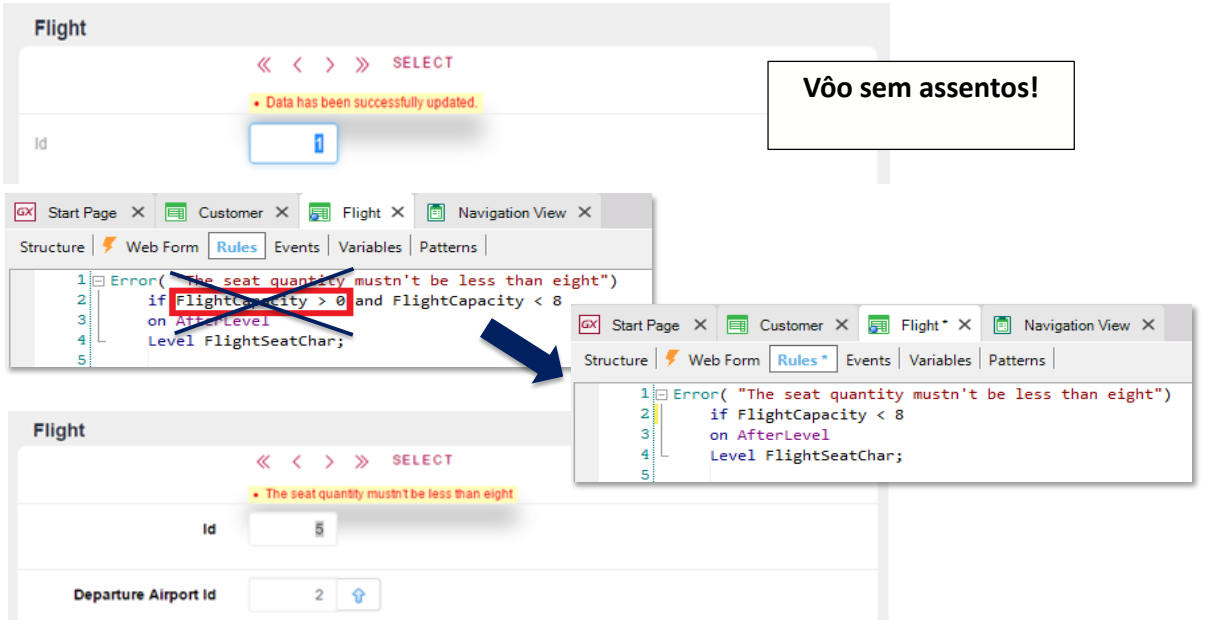
Completemos os 8 lugares exigidos. Digitamos...

3, A, janela
3, B, corredor
4, A, janela, A...e por último...
4, B, corredor

Agora pressionamos Confirmar

E observemos que GeneXus deixou salvar o voo sem problemas.

Exemplo: número de lugares em um voo



Resumindo: **conseguimos nosso propósito retardando o momento em que GeneXus havia escolhido inicialmente para disparar a regra Error.**

O voo 1 havia ficado com 7 lugares, porque havíamos incluindo a regrar Error depois. Enquanto não tentamos gravar este voo, a regrar de Error não será controlada, porque, como aprendemos, será executada depois de CONFIRMAR, quando todos os dados viajam até o servidor. Pressionamos Confirmar e visualizamos a mensagem:

Então, cadastramos um lugar para esse voo:

2-B-Corredor

e salvamos. Agora sim.

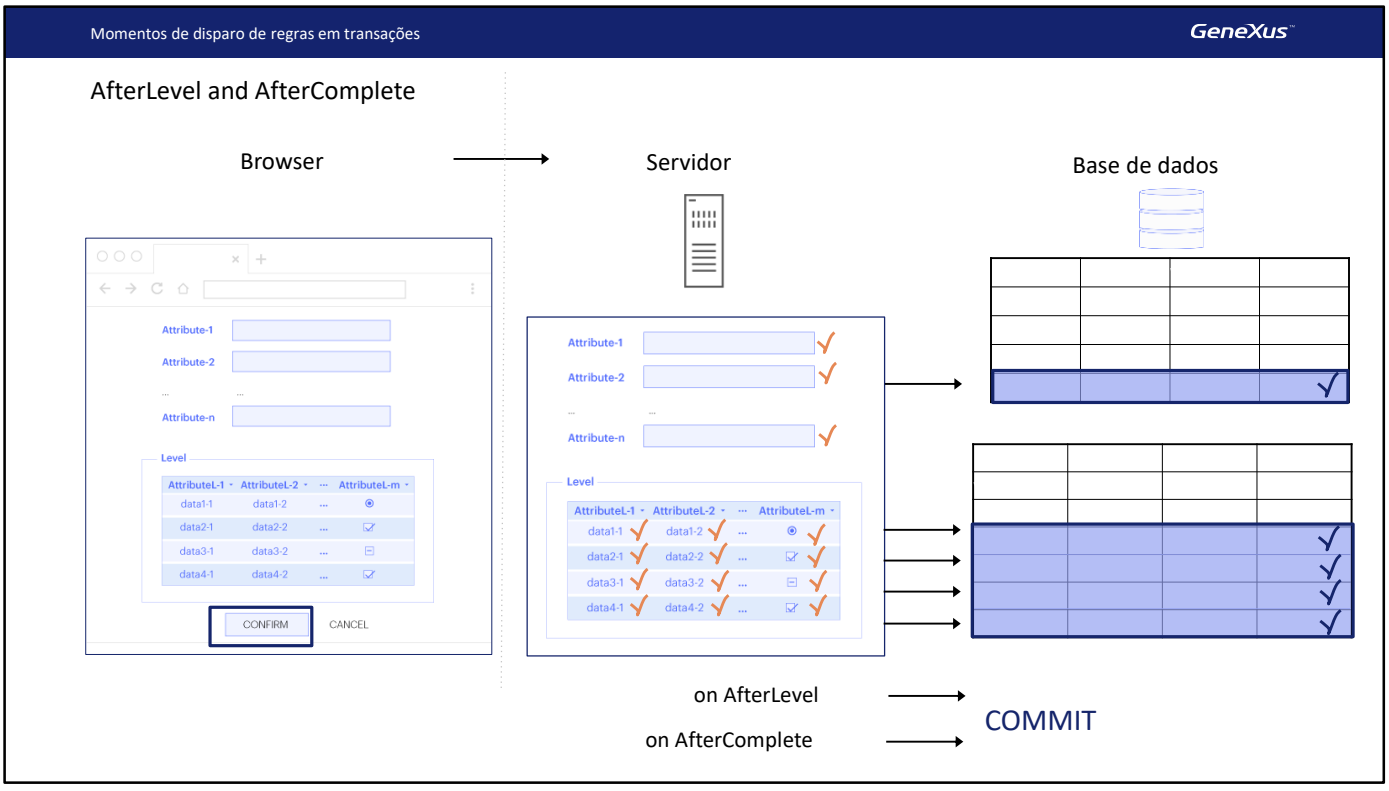
Como último passo, cadastramos um novo voo sem lugares. Permitiu gravar!

Por que? É que temos essa condição:
que inserimos equivocadamente, antes de conhecer que existia o AfterLevel. Portanto, eliminamo-as, e a regra ficará escrita da seguinte maneira:

Apertamos F5, editamos o voo sem lugares e vejamos que agora quando Confirmamos novamente, é controlado que não podemos cadastrar o voo.

Eliminamos o registro, pressionando Delete.

Voltamos ao GeneXus para gravar as alterações que fizemos em nossa KB no GeneXus Server.

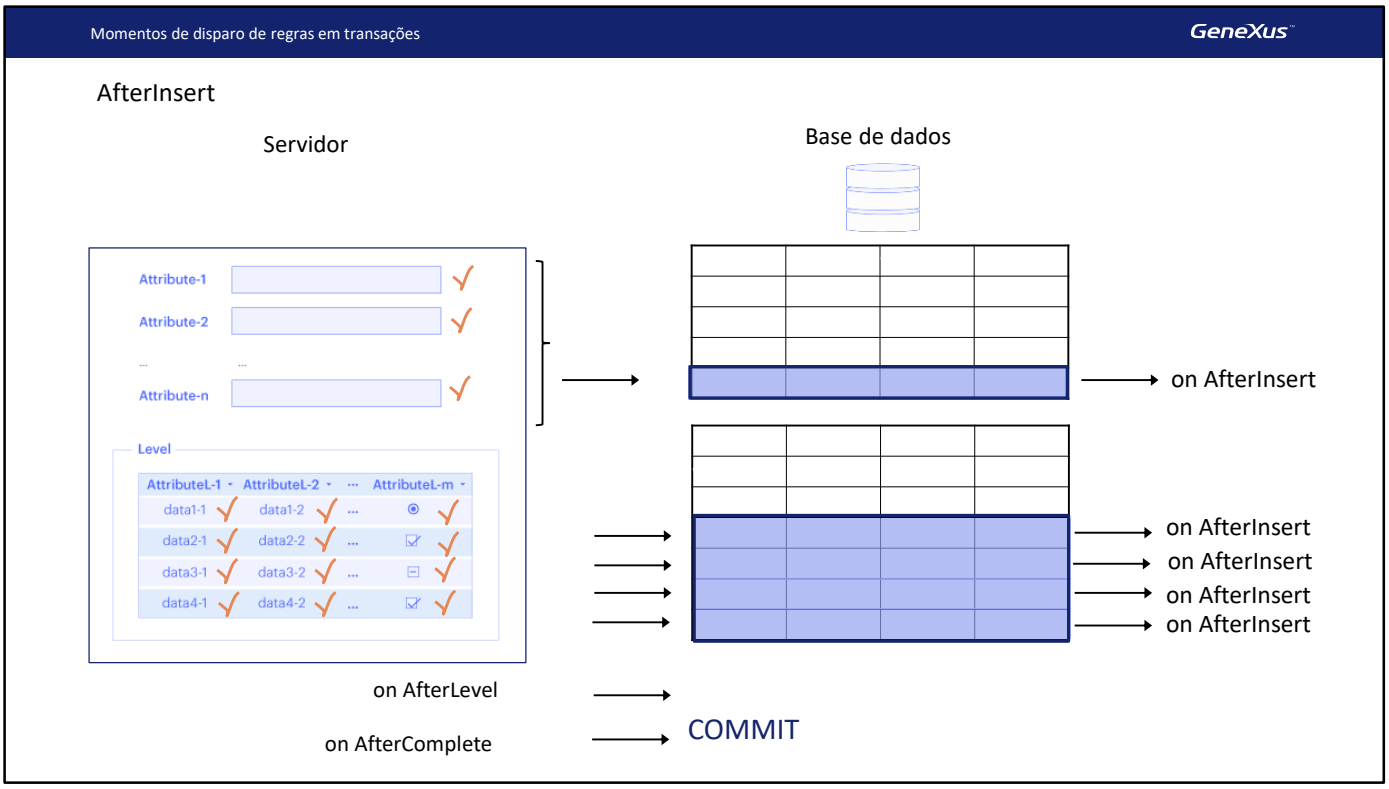


Com esse exemplo, aprendemos que há casos nos quais o momento escolhido por GeneXus para disparar uma regra não é adequado ao nosso interesse, de modo que devemos indicar a GeneXus em que momento queremos que a regra seja executada.

Nesse caso estudamos o momento "**on Afterlevel**",
para indicar que queremos que a regra seja disparada depois de percorrer um nível. Pode servir, por exemplo, para chamar um relatório que imprima dados do voo, já que vimos que no AfterLevel os dados já estão gravados na base de dados, embora se for disparada uma regra Error, os dados não serão gravados.

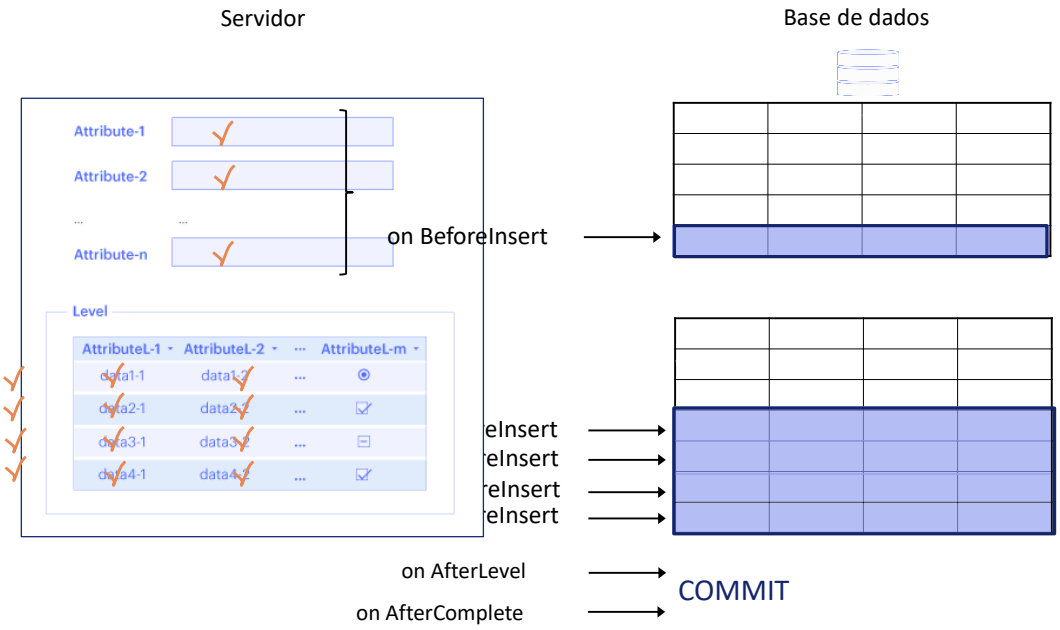
No caso do relatório, seria melhor chamá-lo depois disso, quando estamos seguros que os dados estarão gravados. Isso acontecerá depois de realizado um Commit, comando que logo mencionaremos, cujo efeito é assegurar a gravação dos dados cadastrados.

O momento que ocorre após o commit é o **AfterComplete** e é aí que chamaremos o relatório.



Temos outros momentos como "**on AfterInsert**" para indicar que a regra seja disparada **imediatamente depois** da gravação de cada cabeçalho ou linha:

BeforeInsert



ou "**on BeforeInsert**" se quisermos fazer ou avaliar algo, **imediatamente antes** de que os dados do cabeçalho ou de cada linha sejam armzenados na base de dados.

Observemos que todos esses momentos de disparo começam com o prefixo **on** e sempre são escritos ao finalizar a declaração da regra.

Aqui apenas apresentamos os mais importantes, mas existem mais momentos de disparo disponíveis, que convidamos você a descobrir.

Resumindo

- Às vezes, o momento escolhido pelo GeneXus para acionar uma regra não é a desejada. Nesses casos, devemos indicar especificamente quando queremos que a regra seja acionada.
    - On BeforeInsert
    - On AfterInsert
    - On AfterLevel
    - On AfterComplete
- As regras são condicionadas a esses eventos usando o prefixo "on" escrito no final da regra.

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications