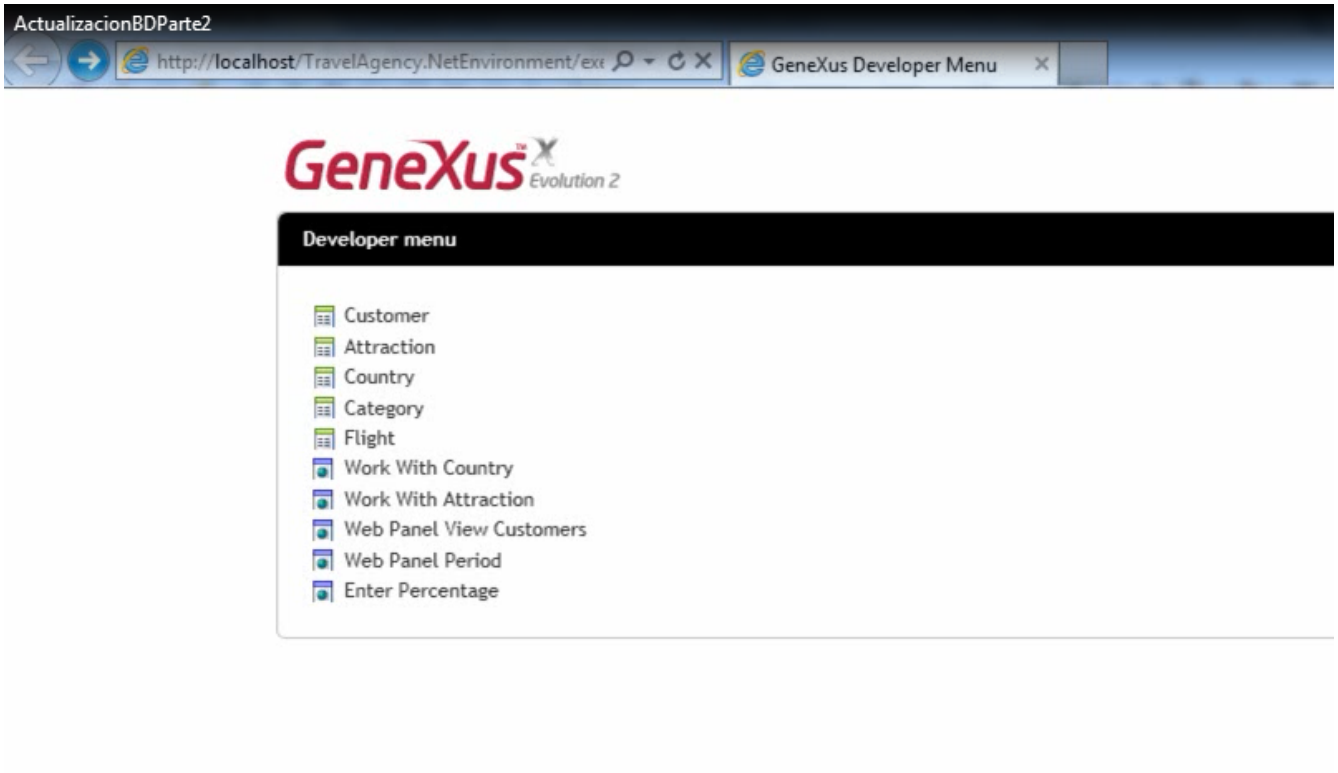
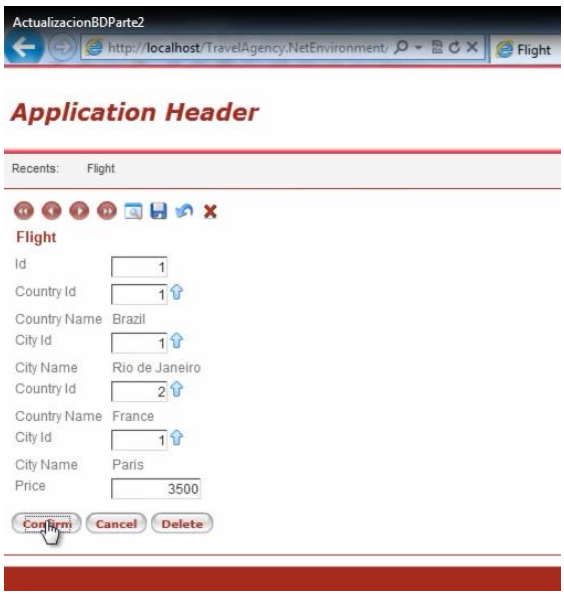
Curso GeneXus - Atualização da Base de Dados -For each , delete , new

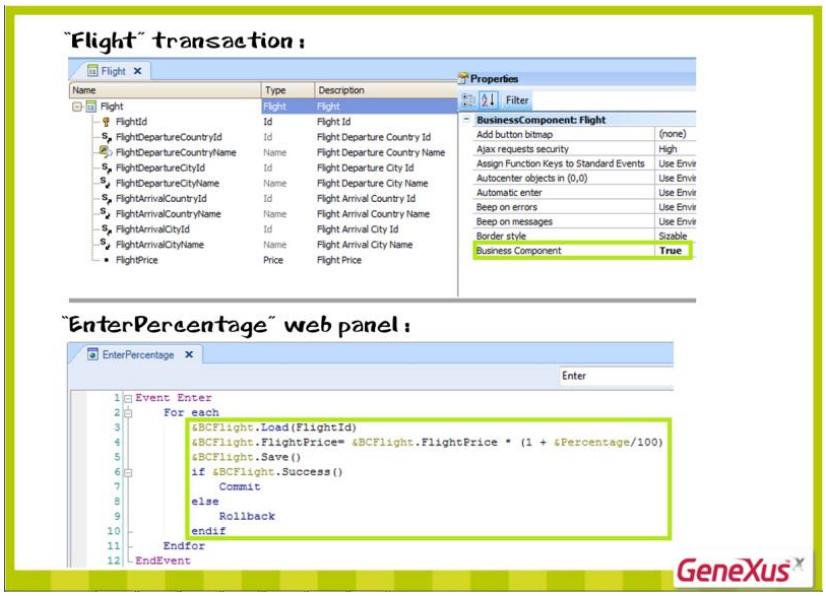


Até o momento, para atualizar os dados do banco de dados, empregamos as transações em suas duas formas de uso:

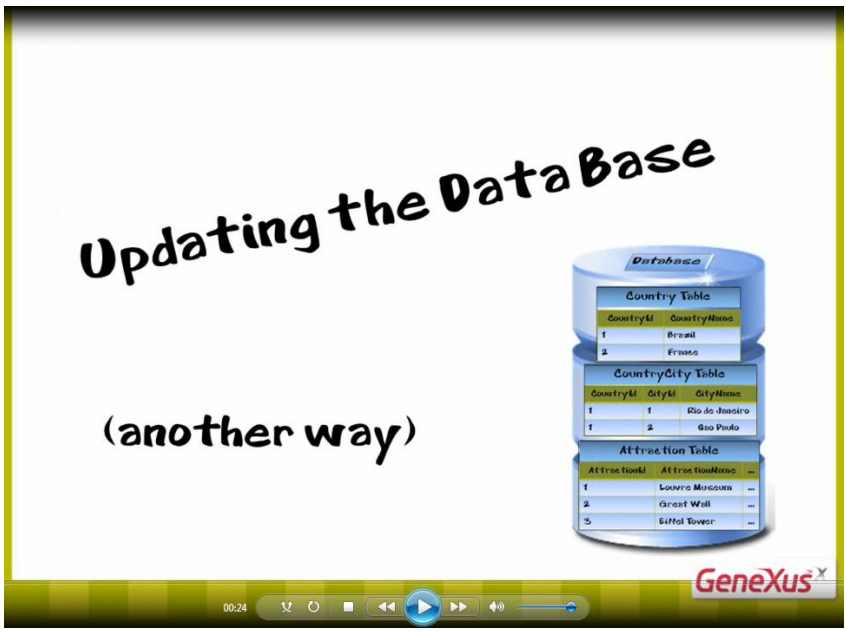
- Executando sua tela e inserindo dados de forma interativa



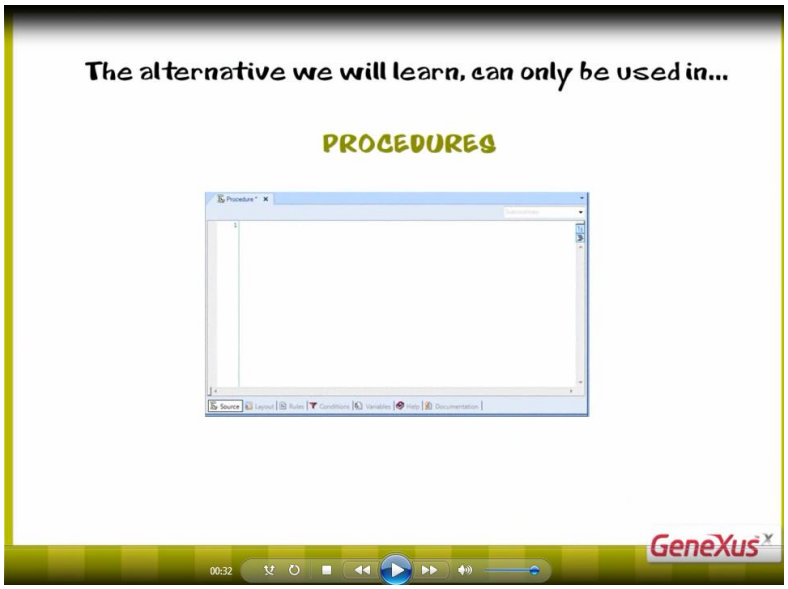
- E executadas como Business Component, através de uma variável, sem usar a tela.



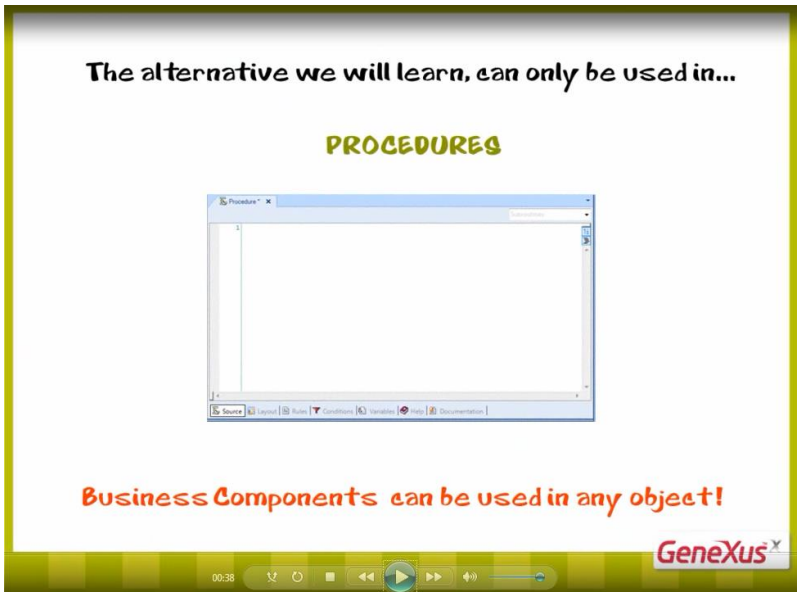
Vamos conhecer agora uma alternativa a mais para realizar inclusões, modificações e exclusões no banco de dados.



Devemos saber que o que veremos **pode ser usado apenas em objetos de tipo procedimento,**

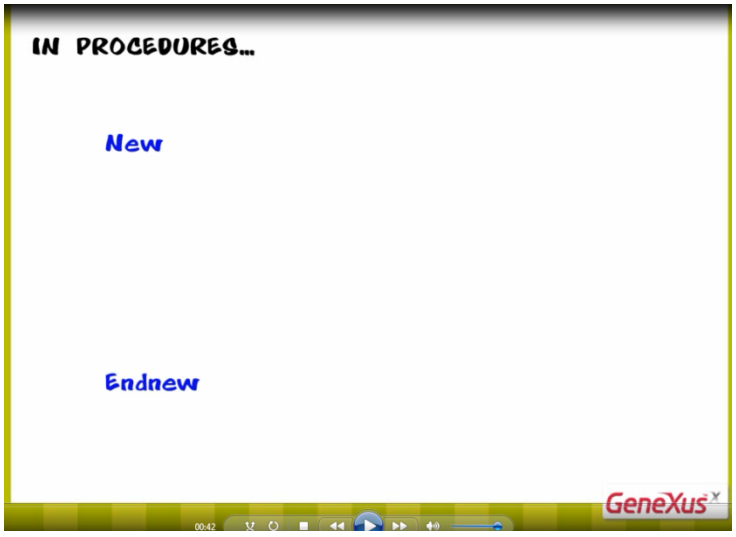


diferente da alternativa que vimos usando Business Component, que podia ser utilizada em qualquer objeto.



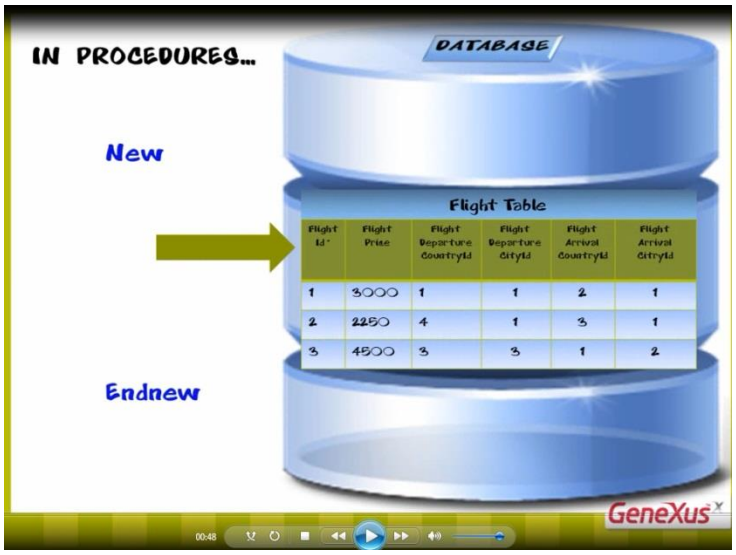
Nos procedimentos,

contamos com um comando chamado New

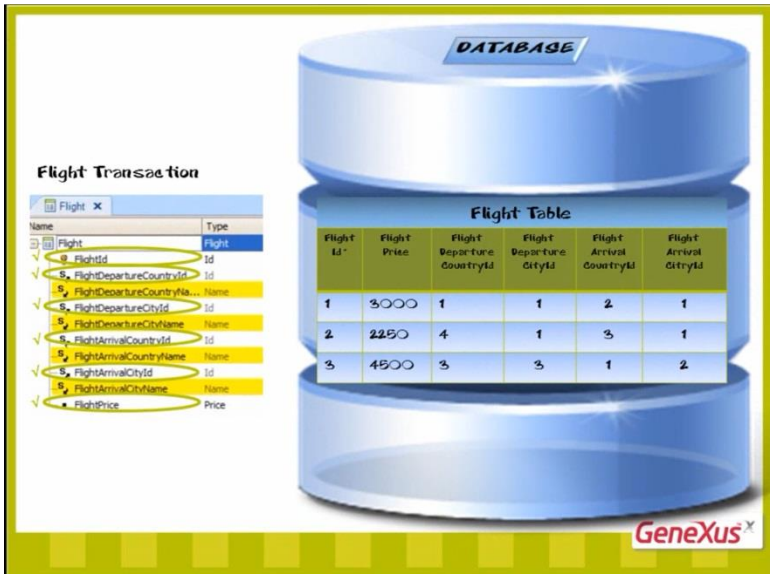


para inserir registros em uma tabela.

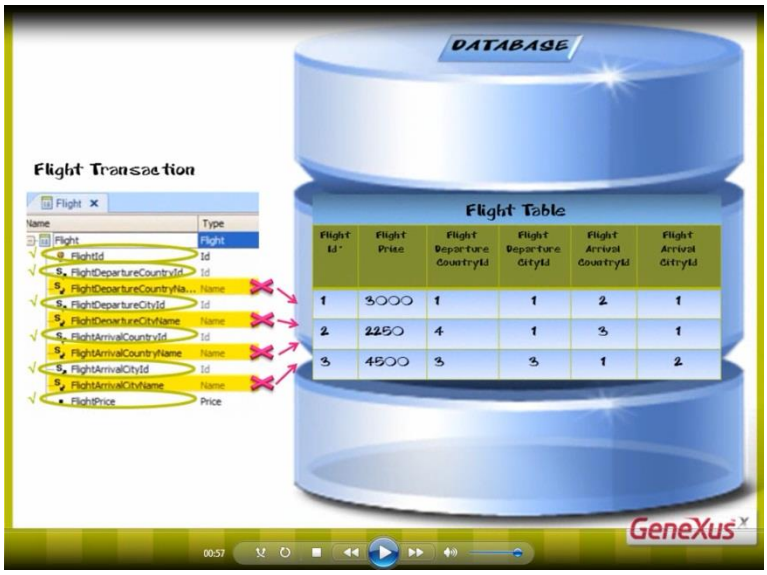
Utilizando esse comando, podemos atribuir valores aos atributos **de uma tabela física**.



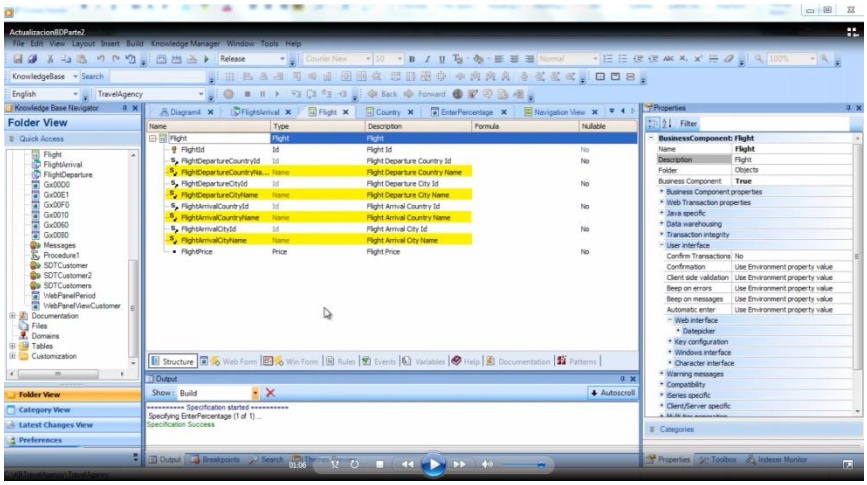
Falamos de uma tabela e não de uma transação



porque nem todos os atributos que estão na estrutura de uma transação estão presentes na tabela física.

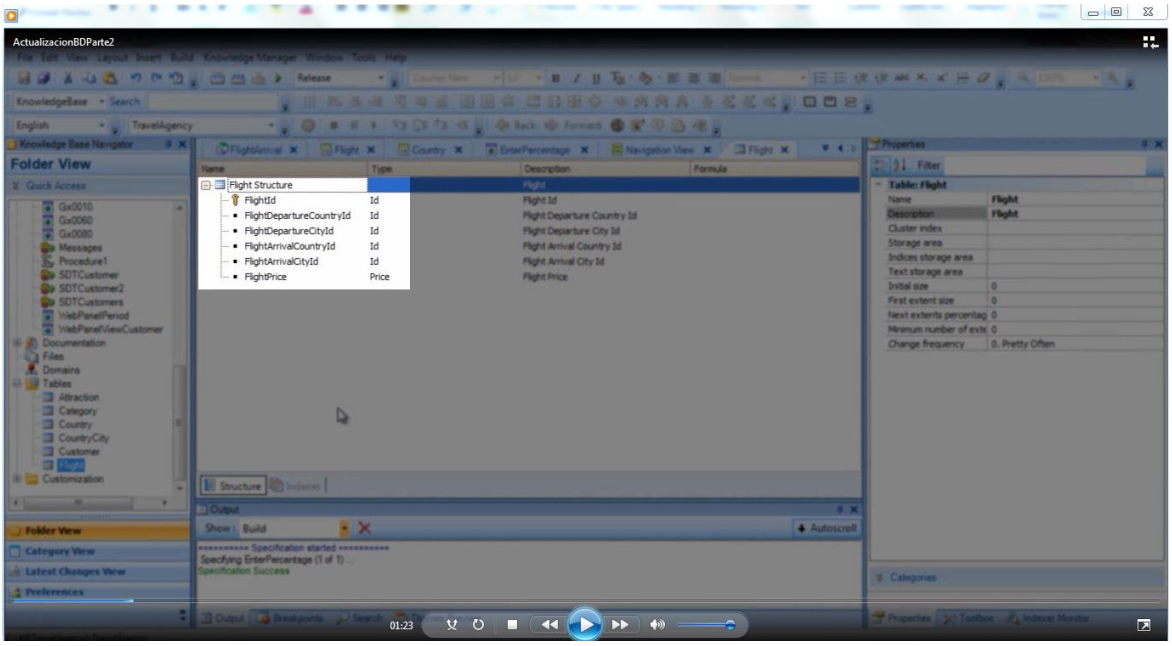


Se quisermos, por exemplo, inserir um voo na transação Flight, há muitos atributos declarados na estrutura



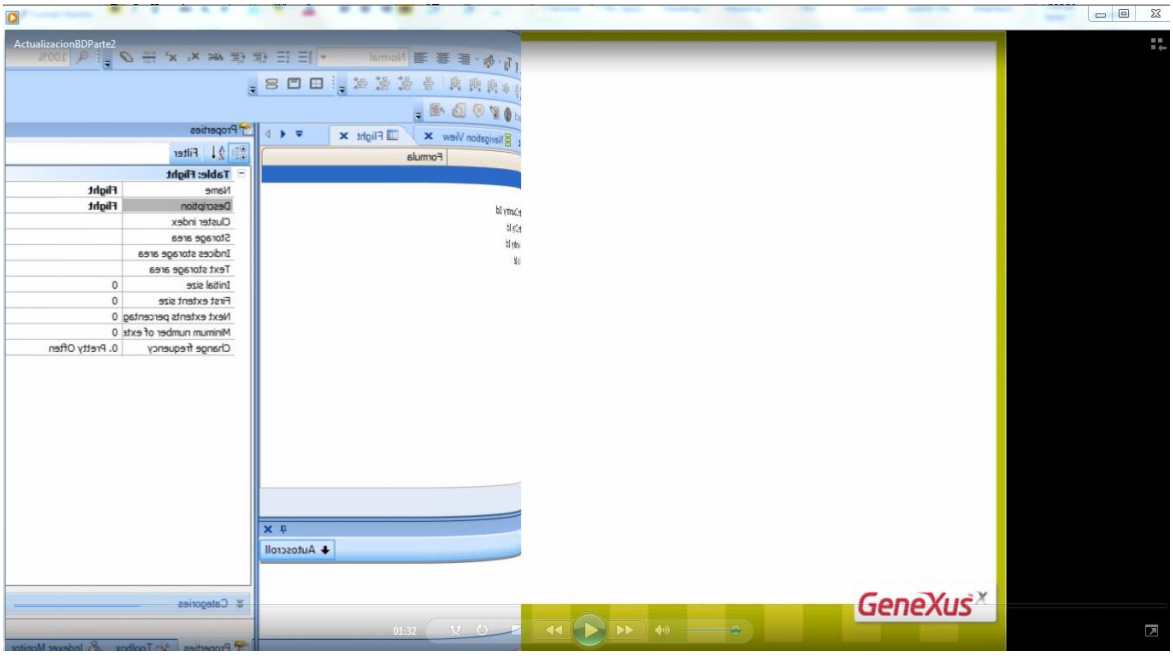
que não estão fisicamente na tabela FLIGHT, mas estão na **tabela estendida da tabela FLIGHT**. Nós os incluímos na estrutura para mostrá-los no form ou para usá-los nas regras.

Se formos ao nó Table e localizarmos a tabela FLIGHT

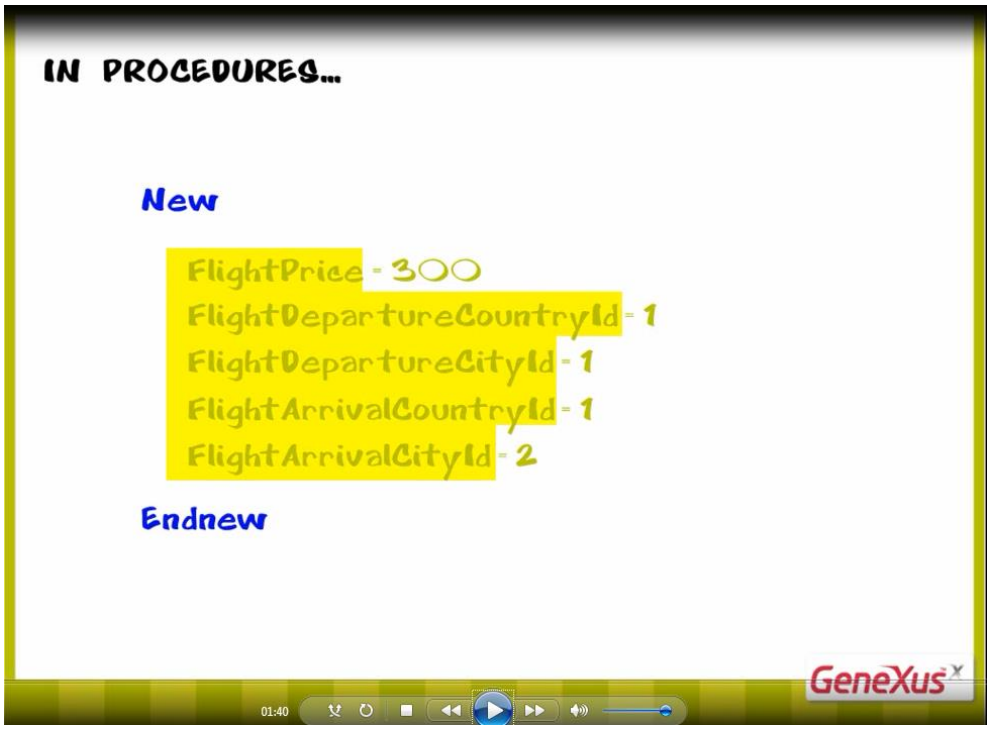


vemos precisamente os atributos **pertencentes à tabela FLIGHT**.

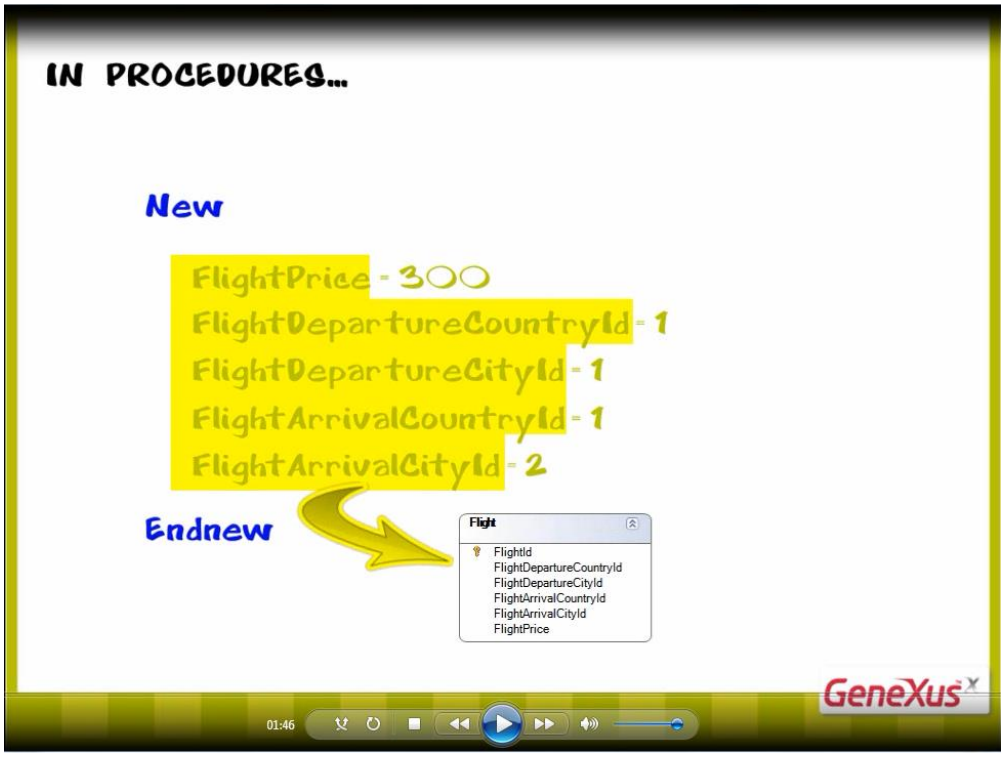
Podemos incluir esses atributos dentro de um comando New e atribuir-lhes valores.



GeneXus determinará qual é a tabela física onde será inserido o registro analisando os atributos que estão **à esquerda do sinal de igual**.

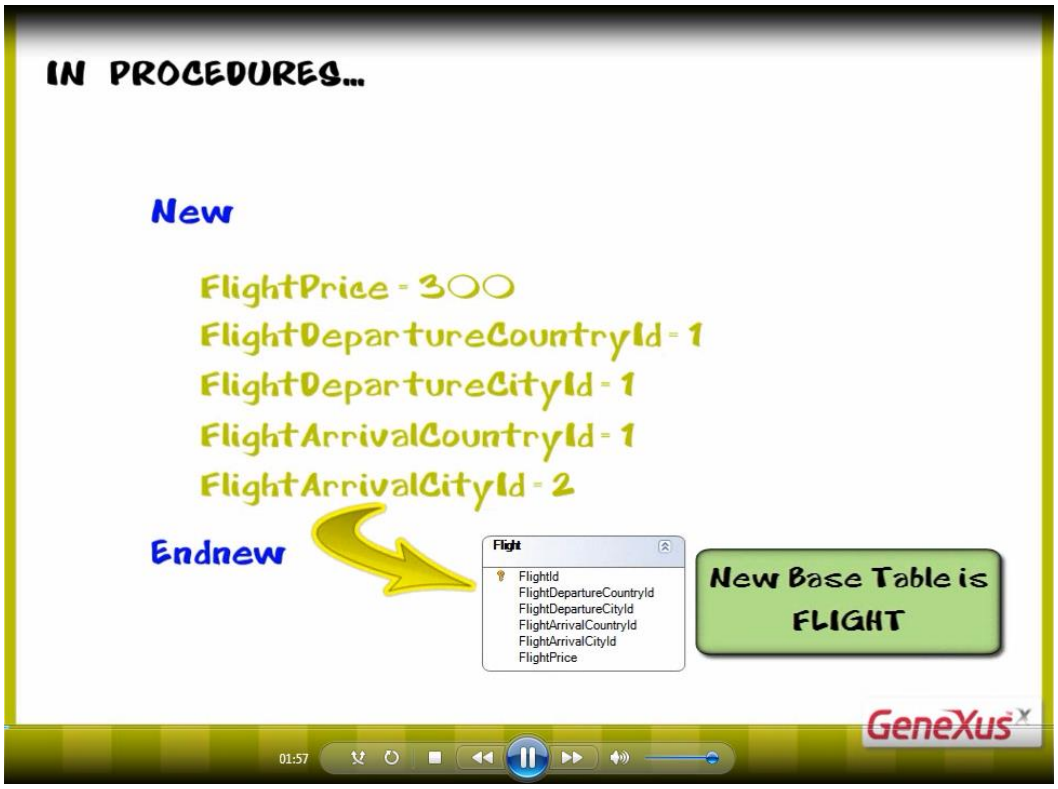


Se todos pertencem a uma mesma tabela física, o registro será inserido em tal tabela;



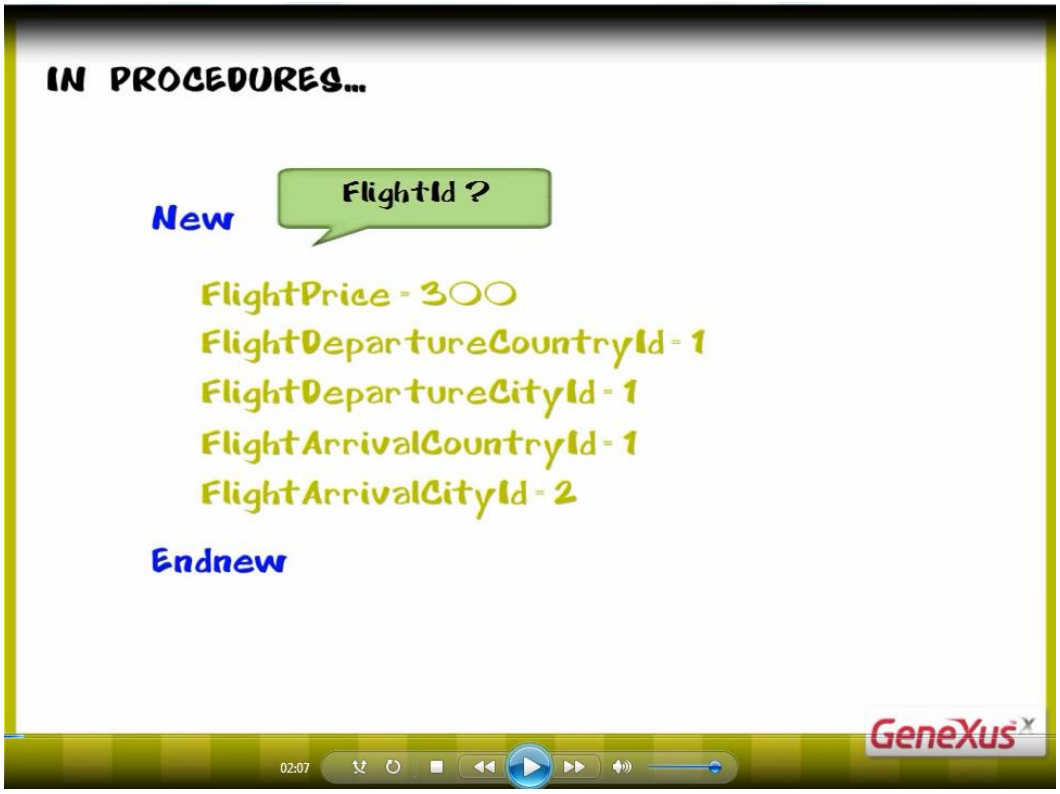
caso contrário, seremos informados que não é possível determinar a tabela na qual se realizará a inclusão.

A tabela que GeneXus encontra chama-se **tabela base do New**



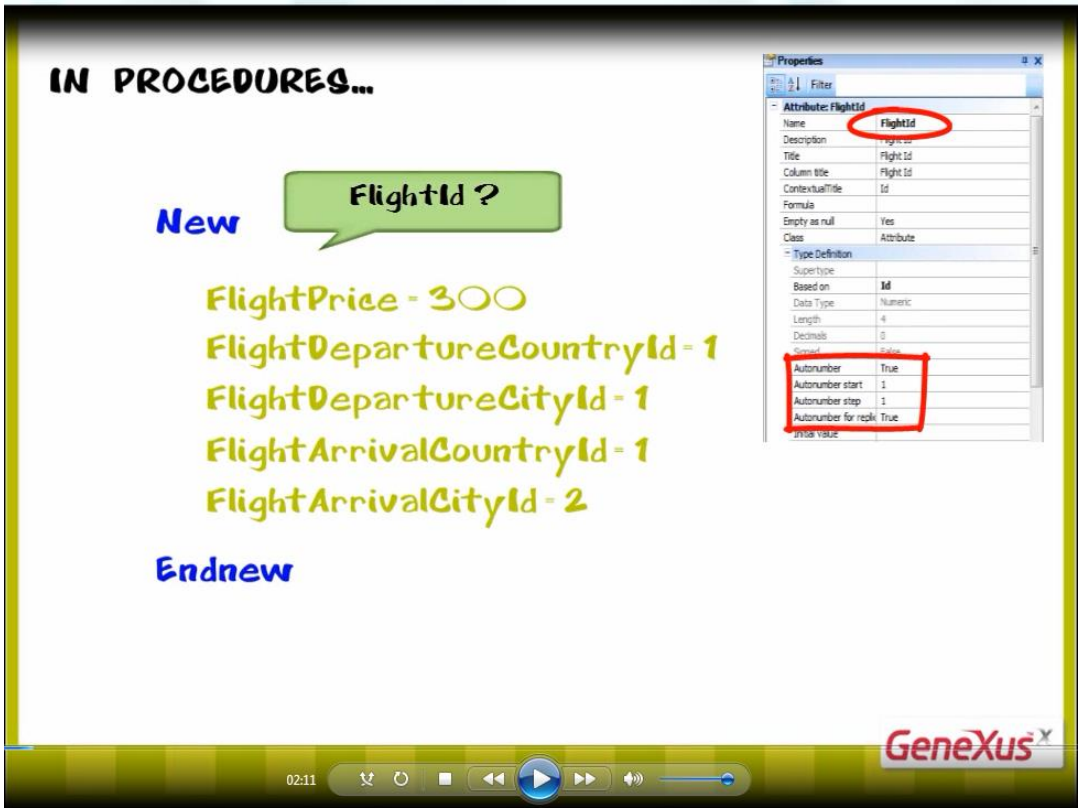
Voltemos à atribuição de valores aos atributos.

Observemos que não atribuímos valor ao atributo chave primária FlightId.

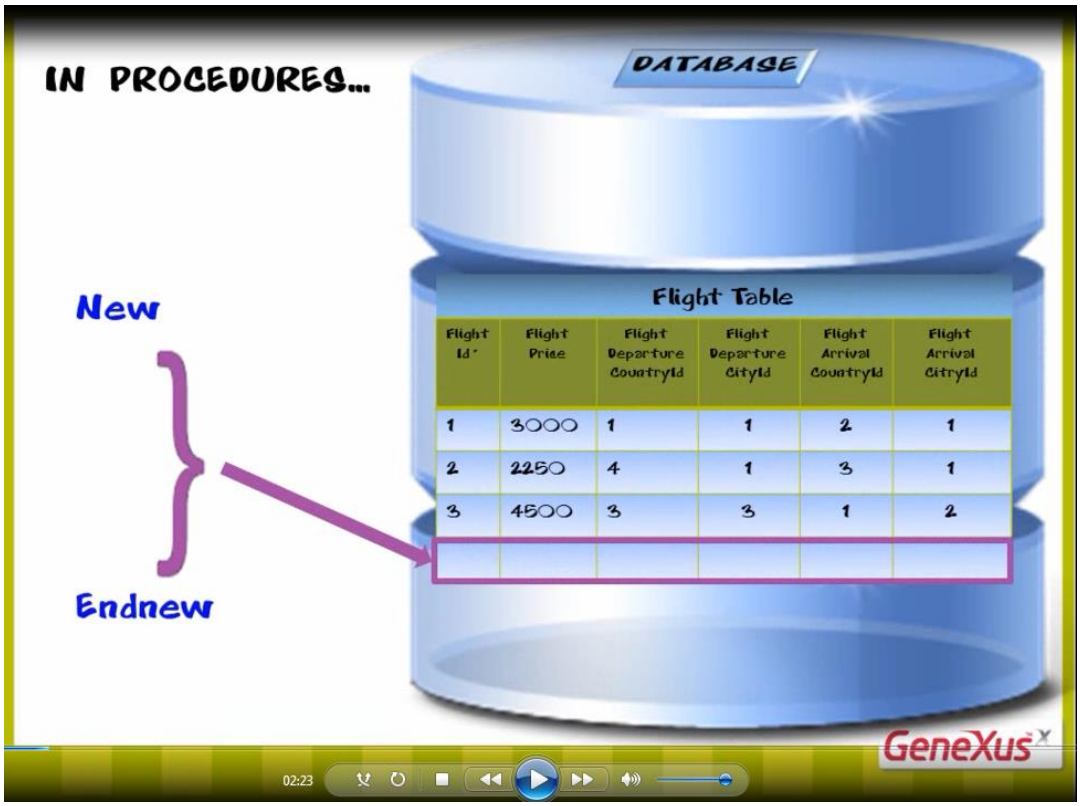


A razão disso é que a propriedade Autonumber de FlightId, está configurada com valor

True, o que faz com que o banco de dados se encarregue de dar valor automaticamente.

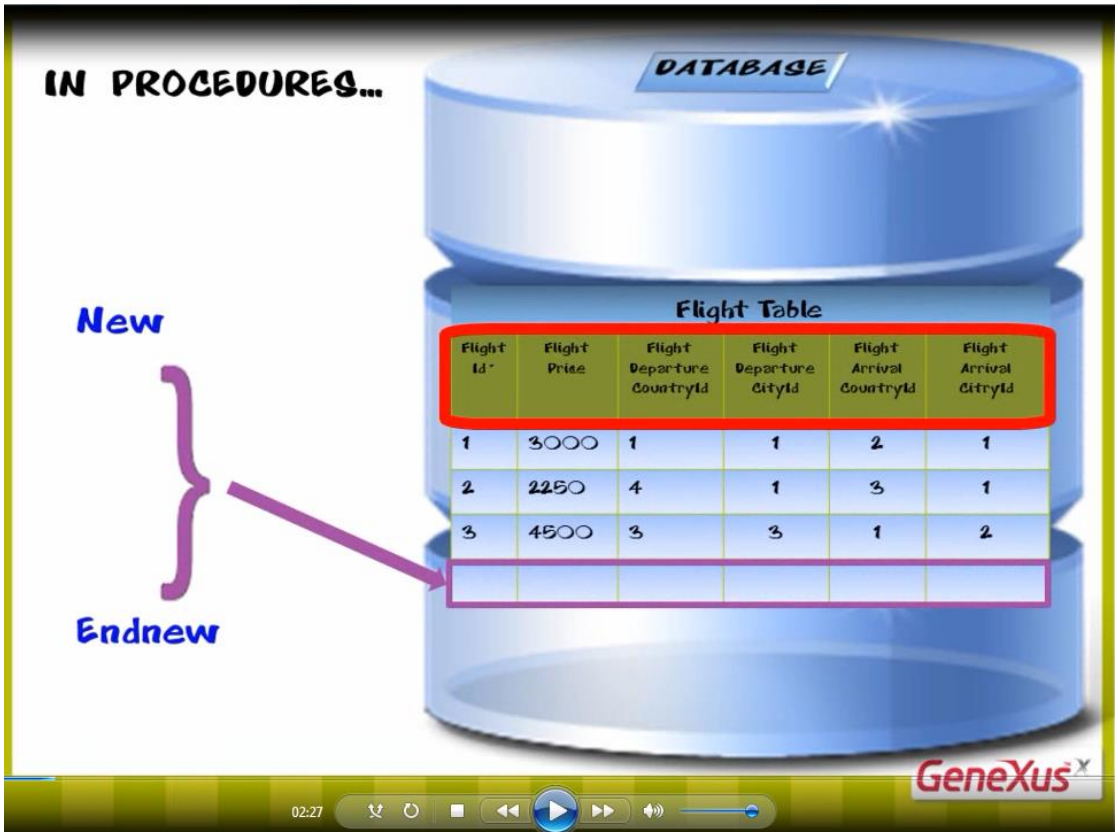


Visto que o comando New insere **um** registro em **uma** tabela,



poderemos atribuir valor somente **aos atributos** que pertencerem a essa **única tabela**

física.

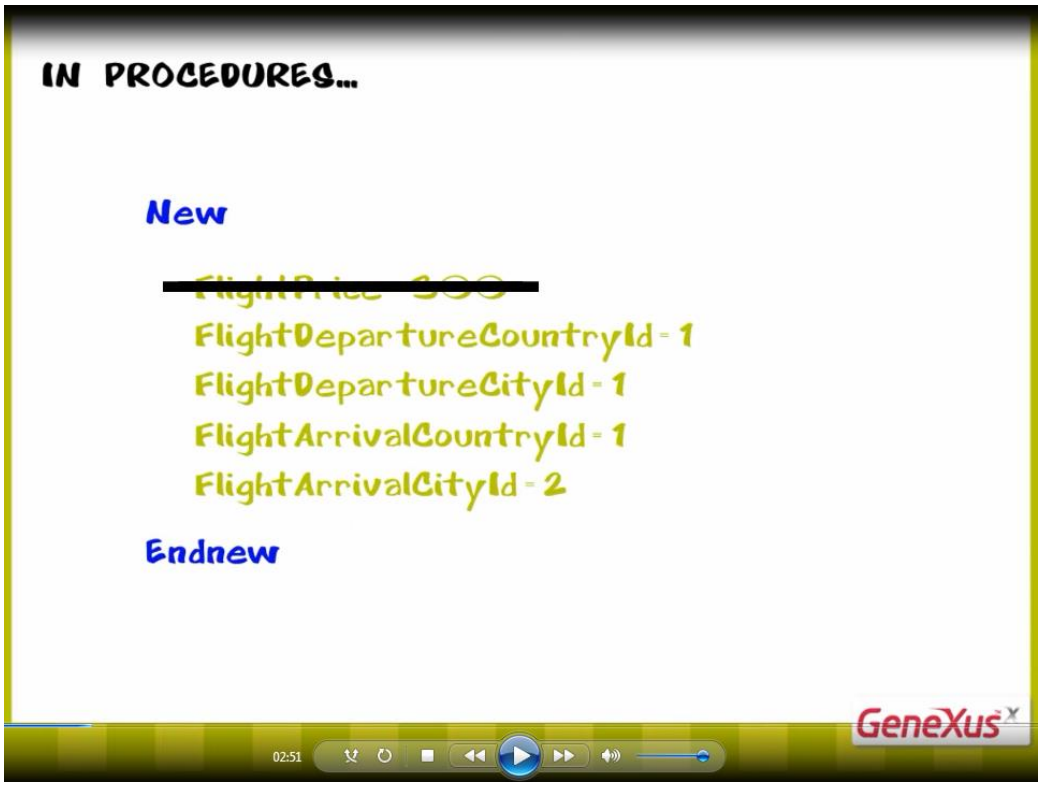


Isso é: não poderemos atribuir valor aos atributos que pertencerem a diferentes tabelas físicas.

Podemos sim omitir a atribuição de um valor a algum atributo da tabela na qual estamos inserindo, seja porque não é necessário (como a FlightId, porque se autonumera),

ou porque queremos deixar algum atributo sem especificar.

Se omitirmos, por exemplo, a atribuição do valor a FlightPrice,

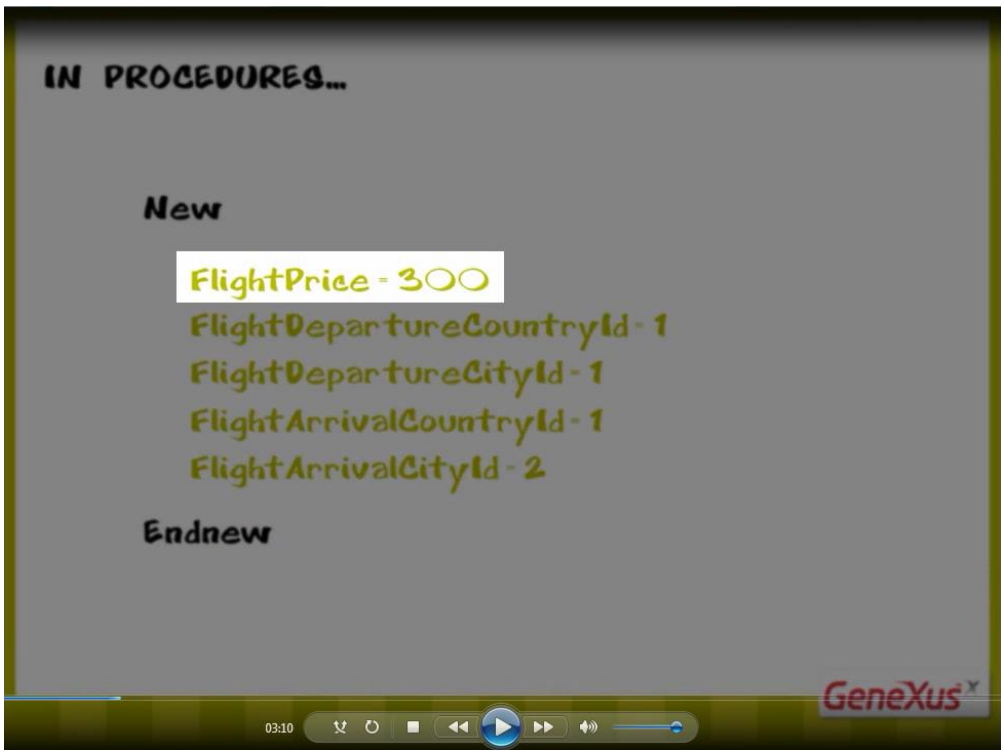


o registro inserido ficará sem preço ou, dito com outras palavras, com preço vazio, sem

especificar.

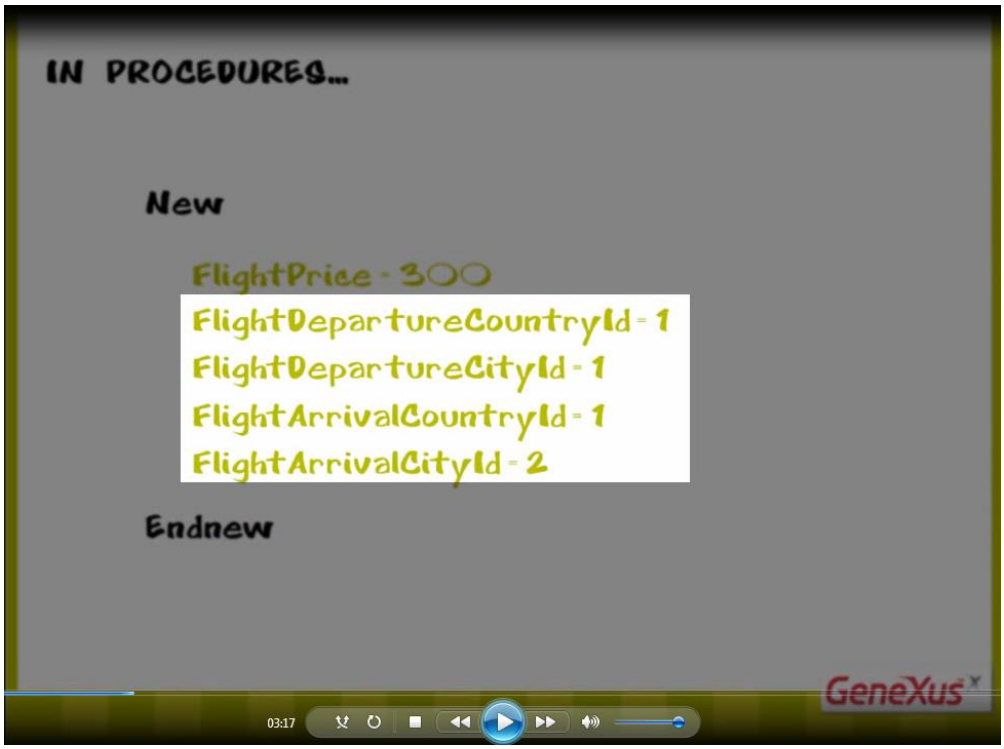
É importante que se saiba que os procedimentos **não controlam se os dados que atribuímos são consistentes.**

Por exemplo, podemos atribuir qualquer valor ao preço do voo,



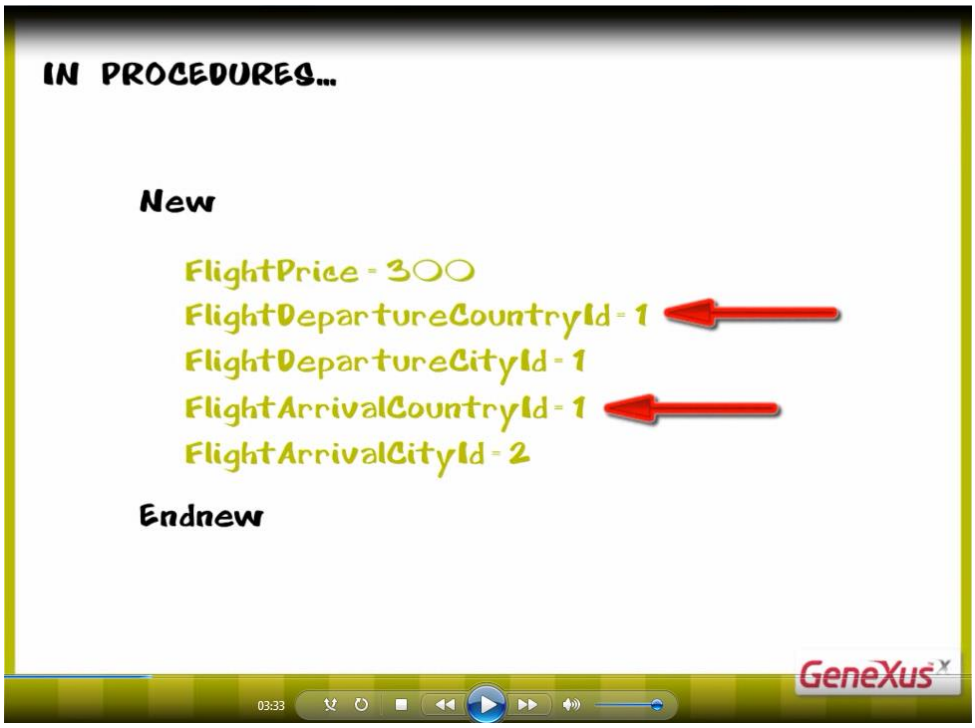
uma vez que não é um dado relacionado com outras tabelas.

Por outro lado, todas as outras atribuições

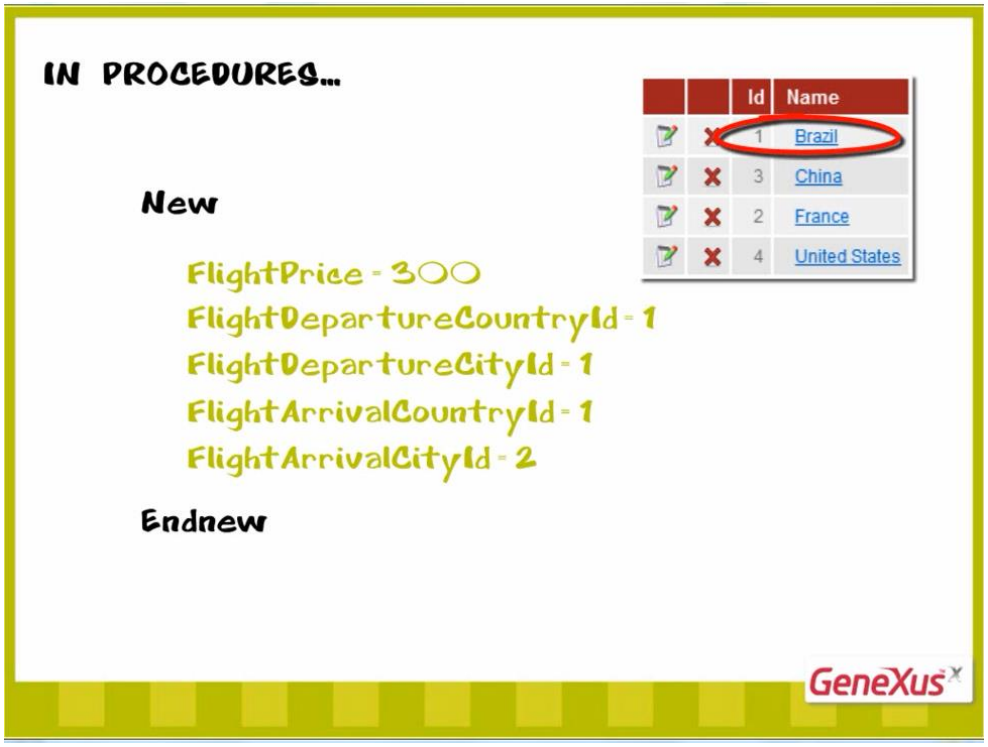


estamos fazendo a identificação de países e cidades... por isso, devemos ser cuidadosos e atribuir valores que existam nas tabelas que armazenam os países e as cidades, respectivamente.

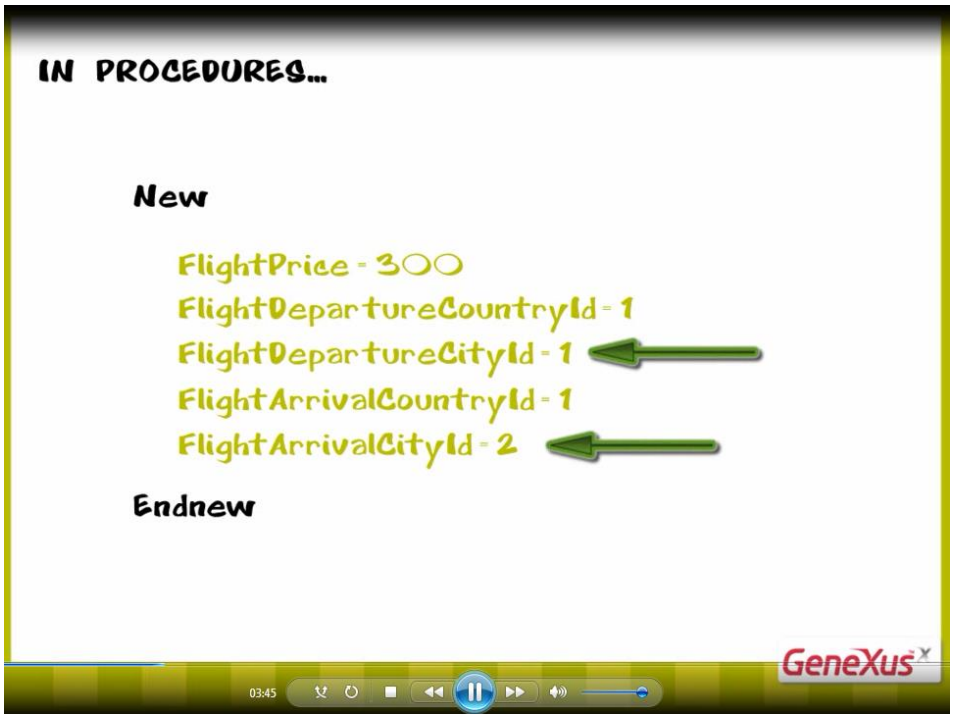
Neste exemplo, atribuímos o valor 1 ao identificador de país em ambos casos,



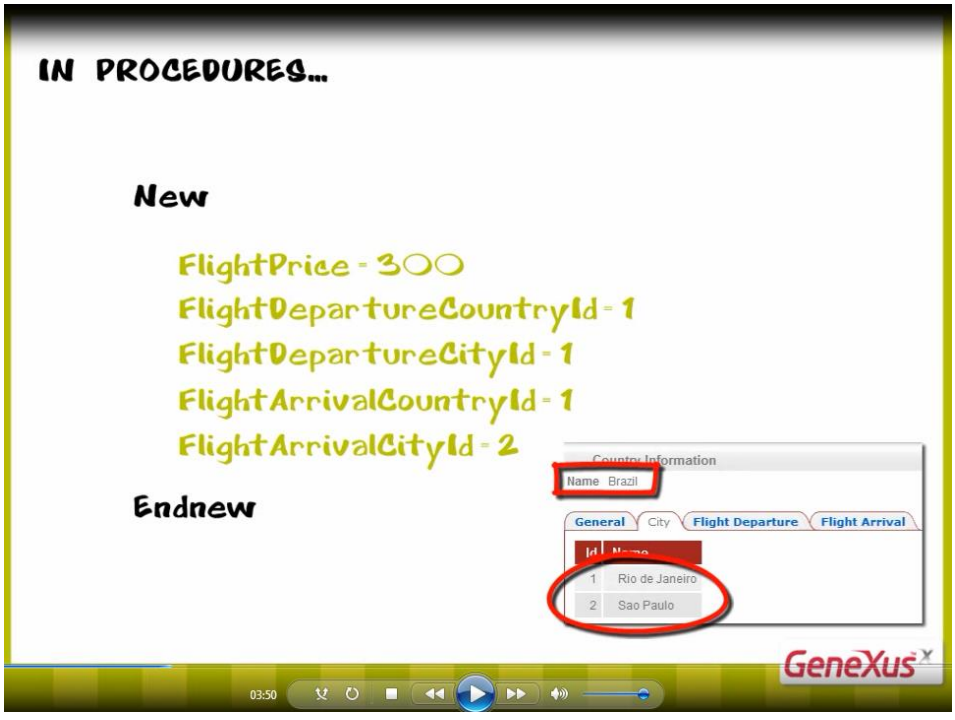
sabendo que, em nossa tabela de países, temos armazenado o país Brasil com identificador: 1



As cidades 1 e 2 que estamos atribuindo

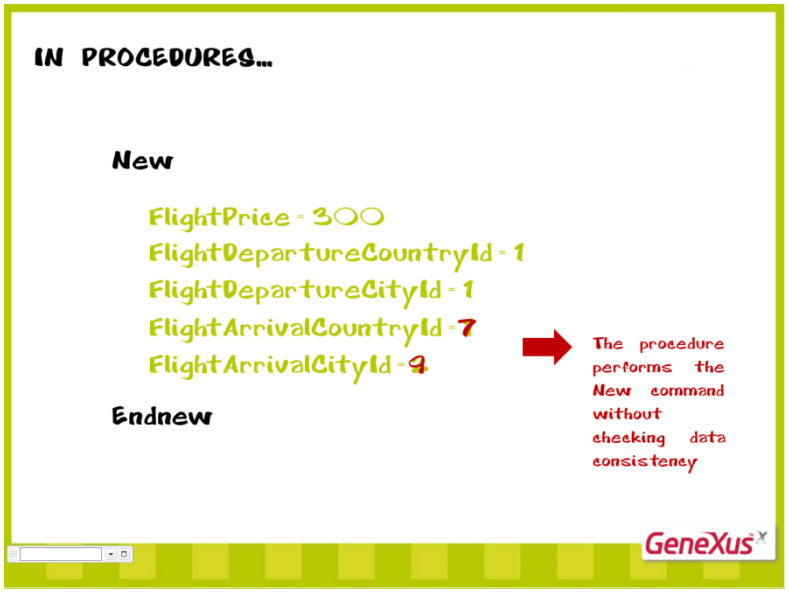


também estão registradas: Rio de Janeiro e São Paulo, respectivamente.

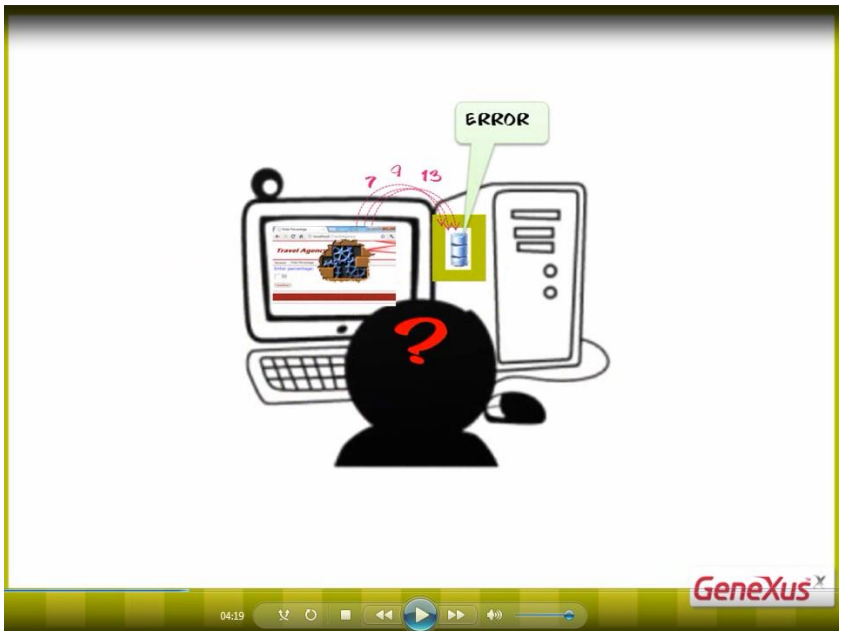


Entretanto, se tivéssemos atribuído um valor de identificador de país ou cidade não registrado em sua correspondente tabela,

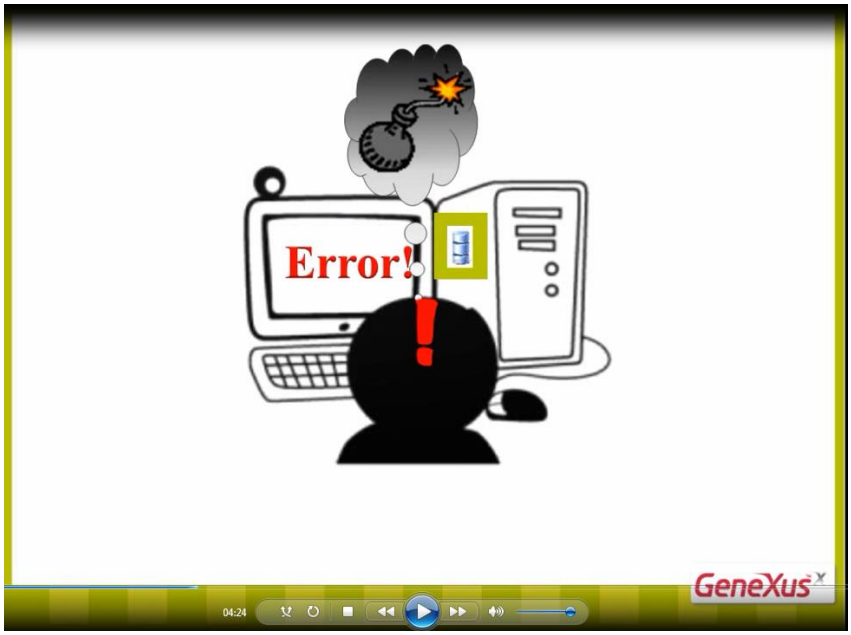
o procedimento não o validaria, porque poderíamos estar inserindo dados inconsistentes.



Sendo assim, como os bancos de dados controlam a consistência dos dados inter-relacionados, quando o usuário executa o aplicativo e tenta atribuir um valor não consistente, o banco de dados rejeita a operação e a gravação inconsistente não será concluída.

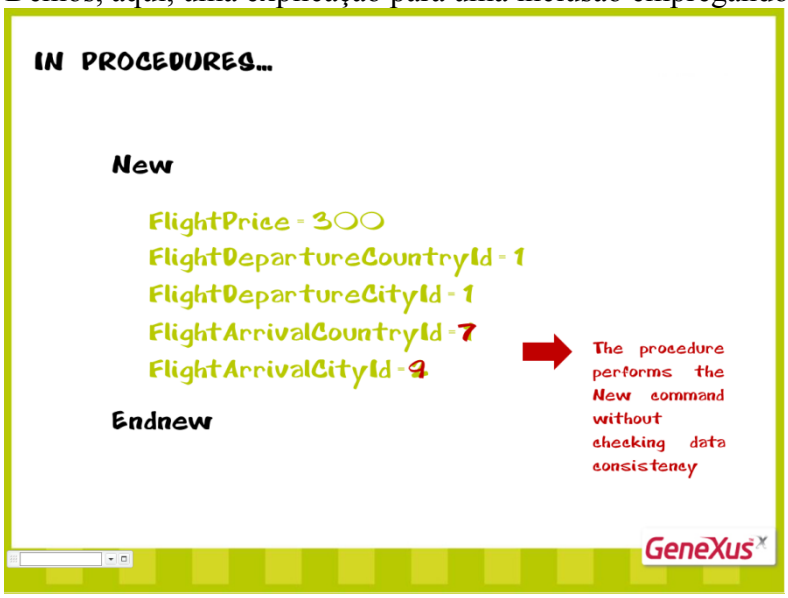


O programa cancelará sua execução, mas isso não é muito amigável para o usuário.

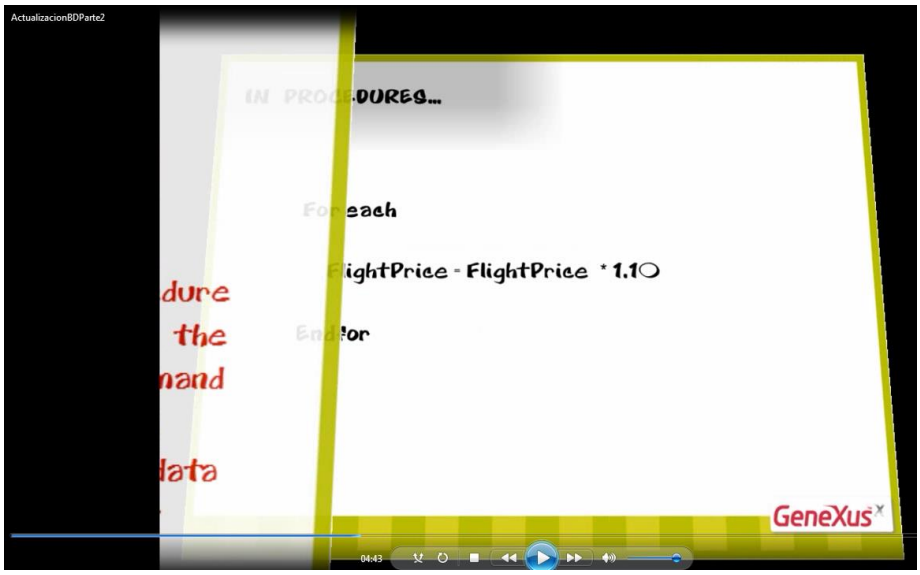


Portanto, se usarmos os procedimentos para atualizar o banco de dados, **será nossa responsabilidade atribuir dados válidos e bem relacionados**.

Demos, aqui, uma explicação para uma inclusão empregando o comando New,

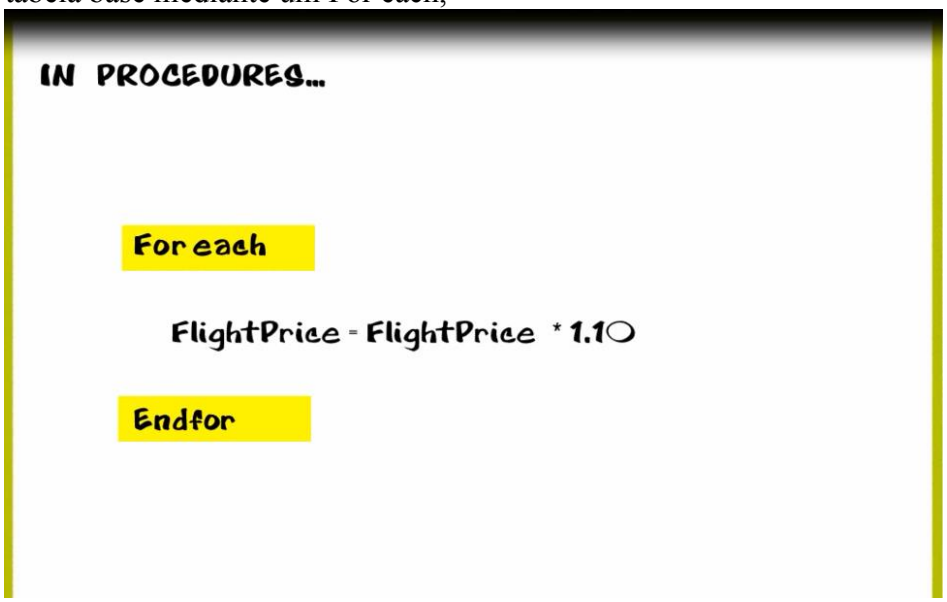


mas também é preciso considerá-la ao atualizar dados ou excluir registros.

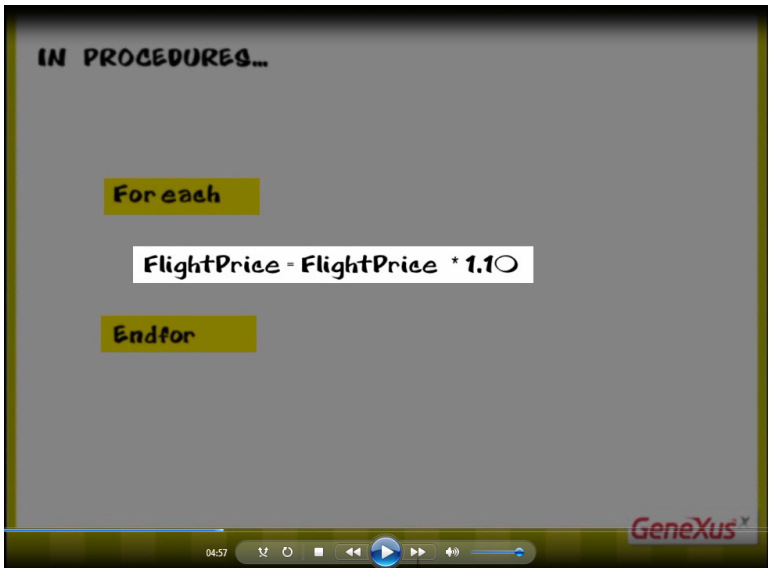


Vejamos agora como podemos atualizar um valor existente no banco de dados.

Para mudar um valor armazenado em um atributo por outro valor, navegamos pela tabela base mediante um For each,

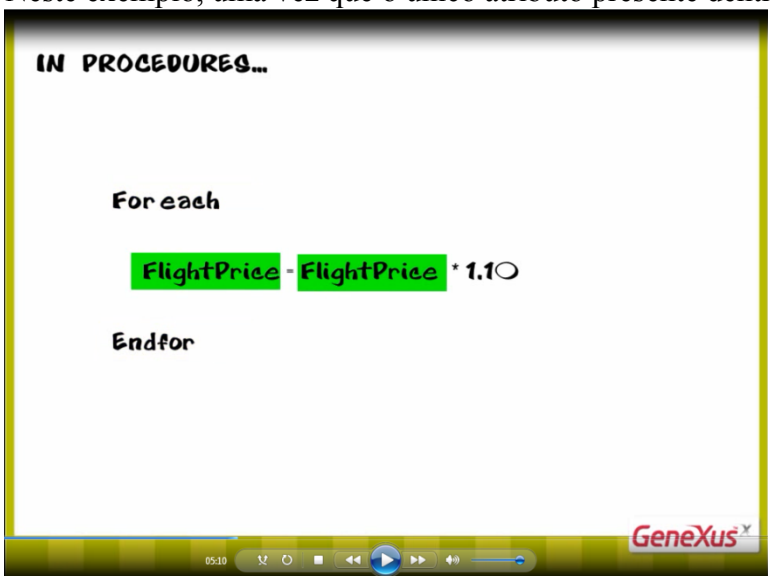


e mediante uma atribuição, damos o novo valor.

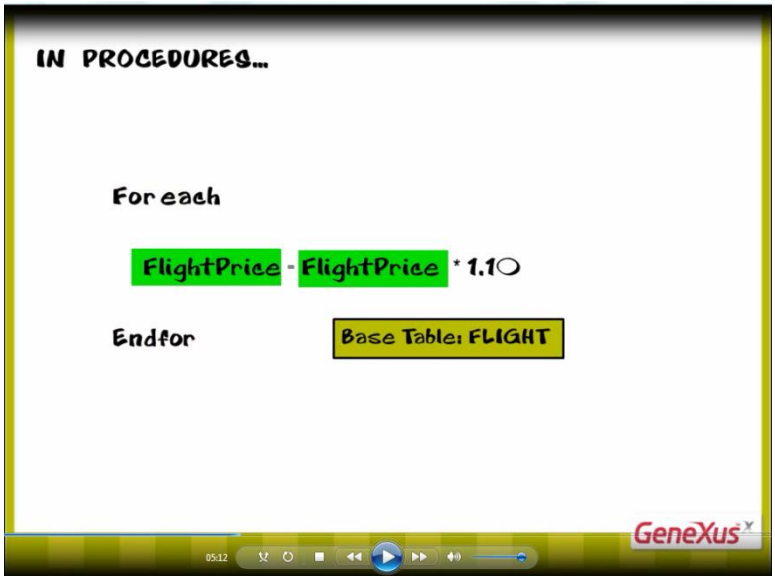


Podemos realizar atribuições aos atributos da tabela base que estamos navegando e aos da tabela estendida.

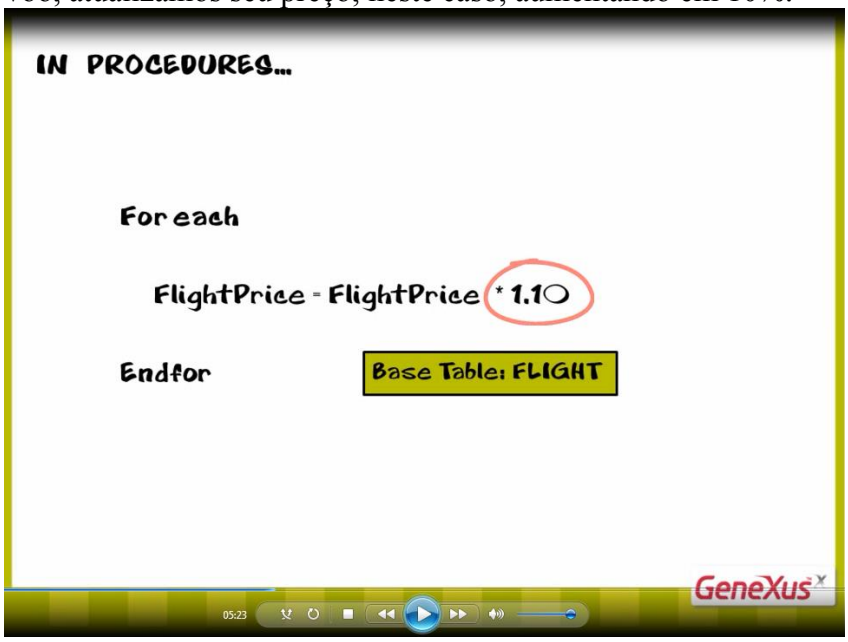
Neste exemplo, uma vez que o único atributo presente dentro do For each é FlightPrice,



a tabela base que GeneXus navegará é: FLIGHT



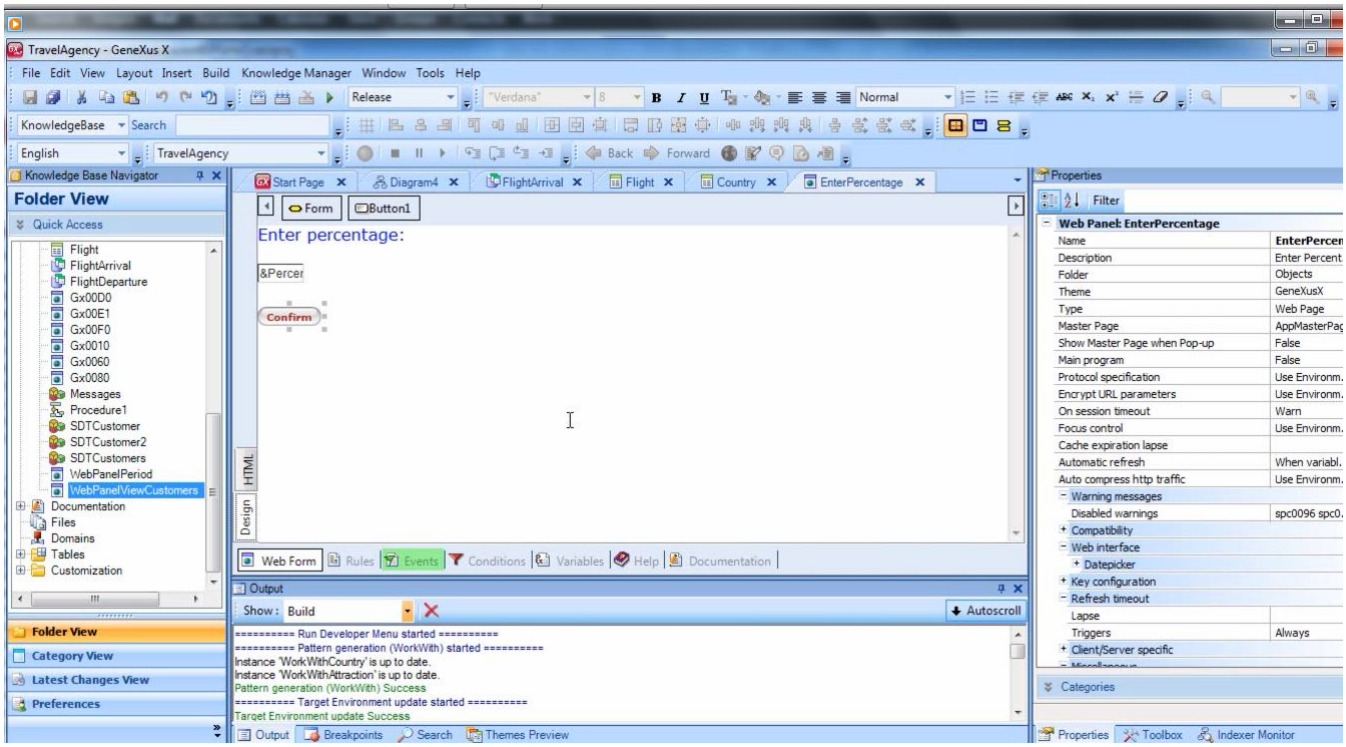
Como não há filtros definidos, se navegará por todos os registros da tabela e, para cada voo, atualizamos seu preço; neste caso, aumentando em 10%.



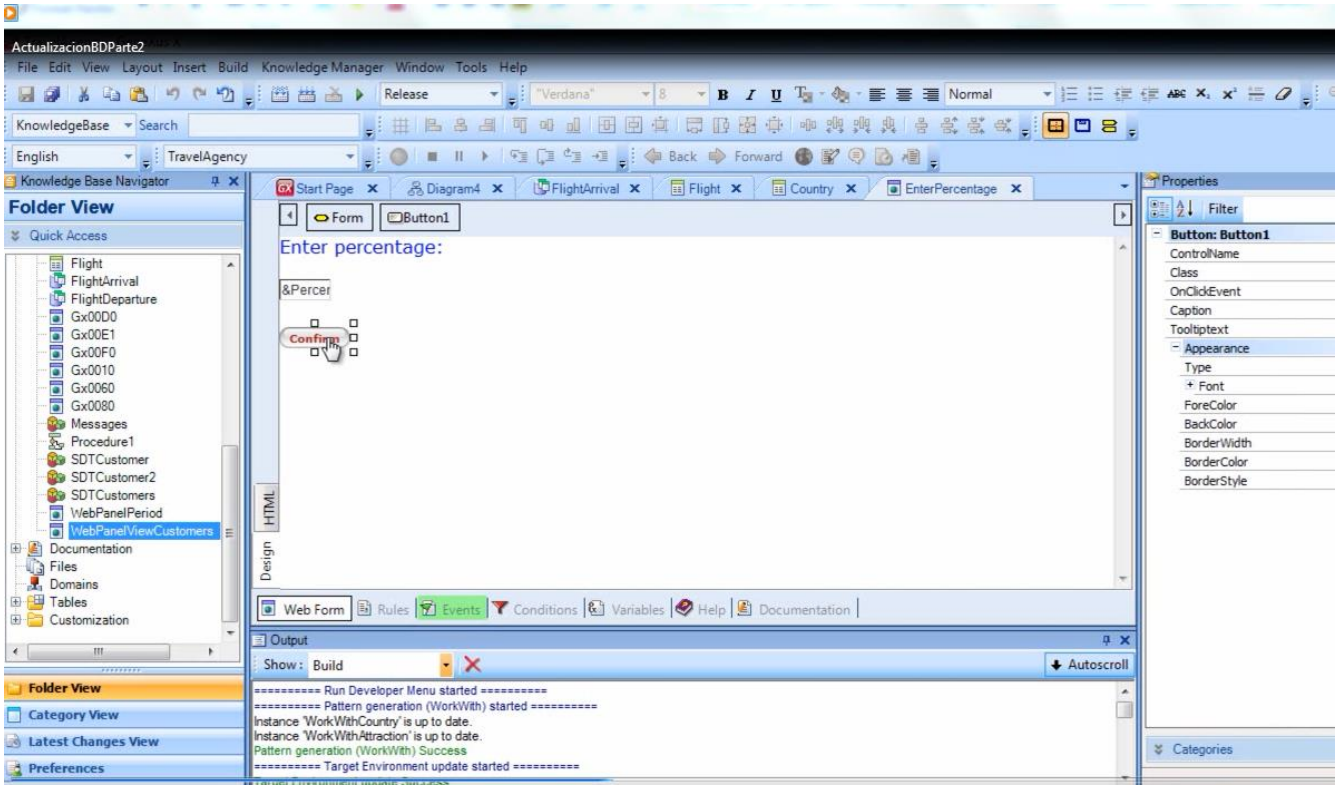
Vamos agora a GeneXus, para por em prática o que vimos.

Vamos resolver a mesma funcionalidade que havíamos implementado empregando o conceito de business component e poderemos comparar ambas as soluções.

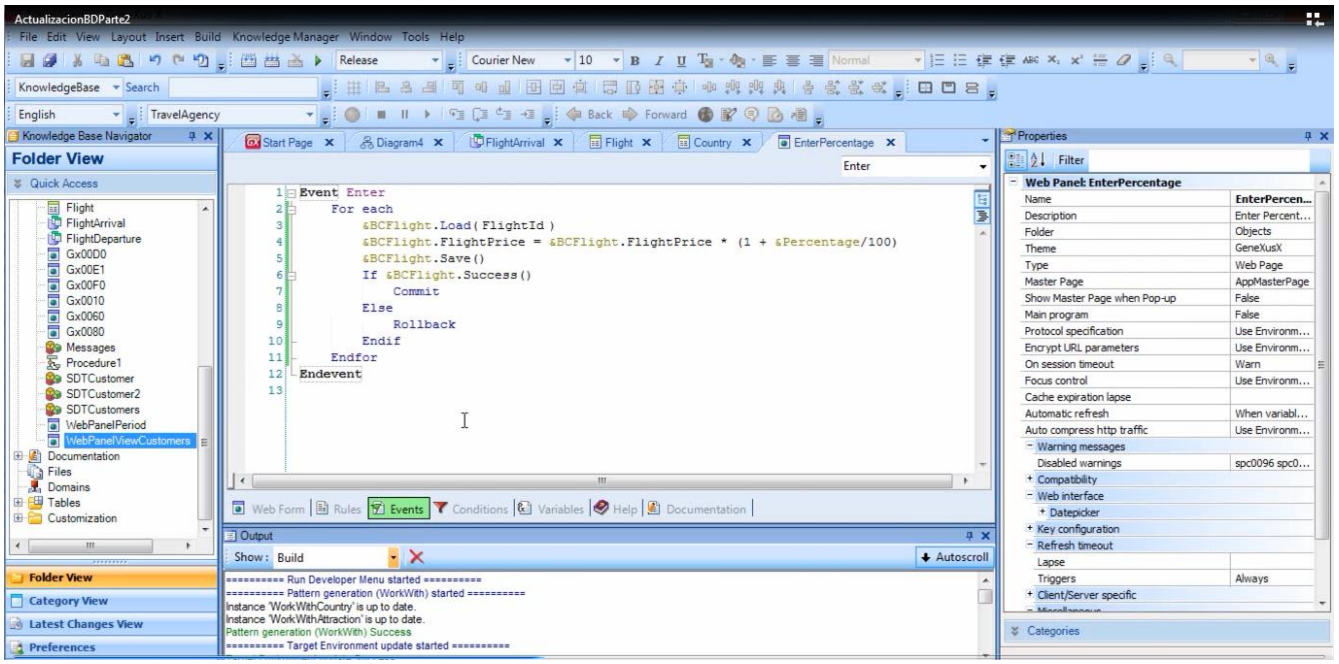
Lembremos que tínhamos um web panel chamado “EnterPercentage”



no qual o usuário da agência de viagens podia digitar uma porcentagem e, ao clicar sobre o botão confirmar,

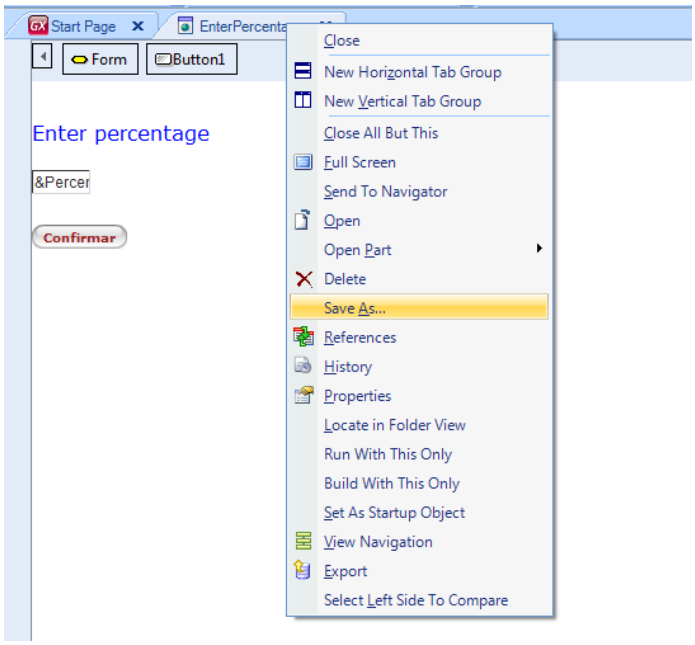


executava-se **este código**



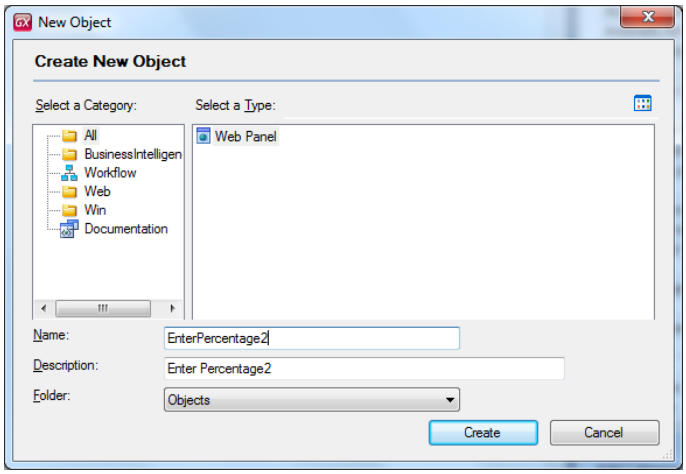
para navegar por todos os voos e incrementar o preço de cada voo.

Vamos clicar com o botão direito do mouse sobre a guia com o nome do objeto

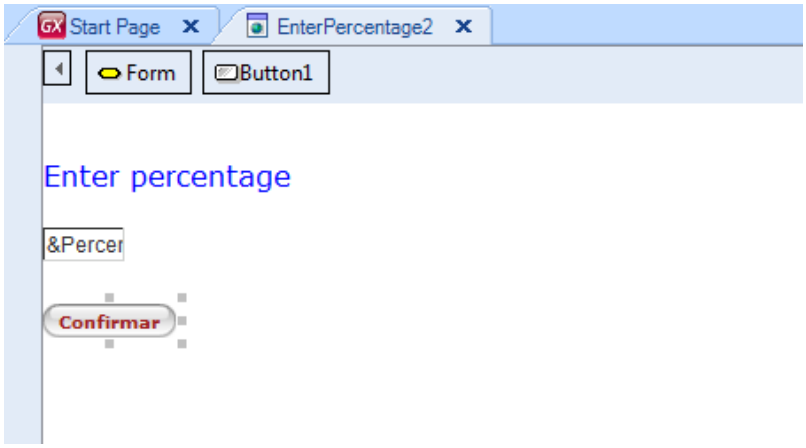


e escolhemos “Save as” para obter uma cópia do objeto com outro nome.

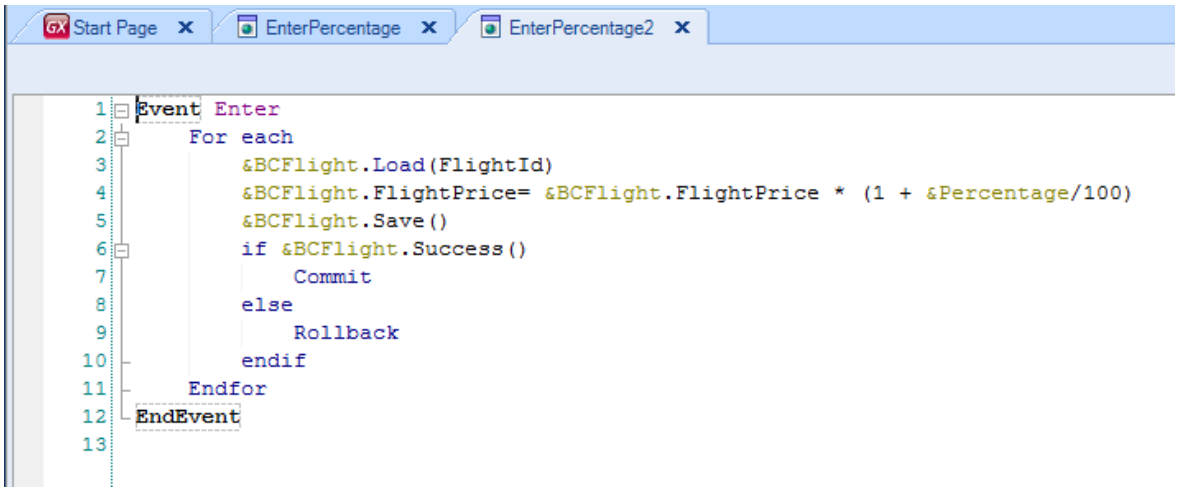
O nomeamos “EnterPercentage2”



Aqui ficou o novo web panel, até agora idêntico ao anterior.

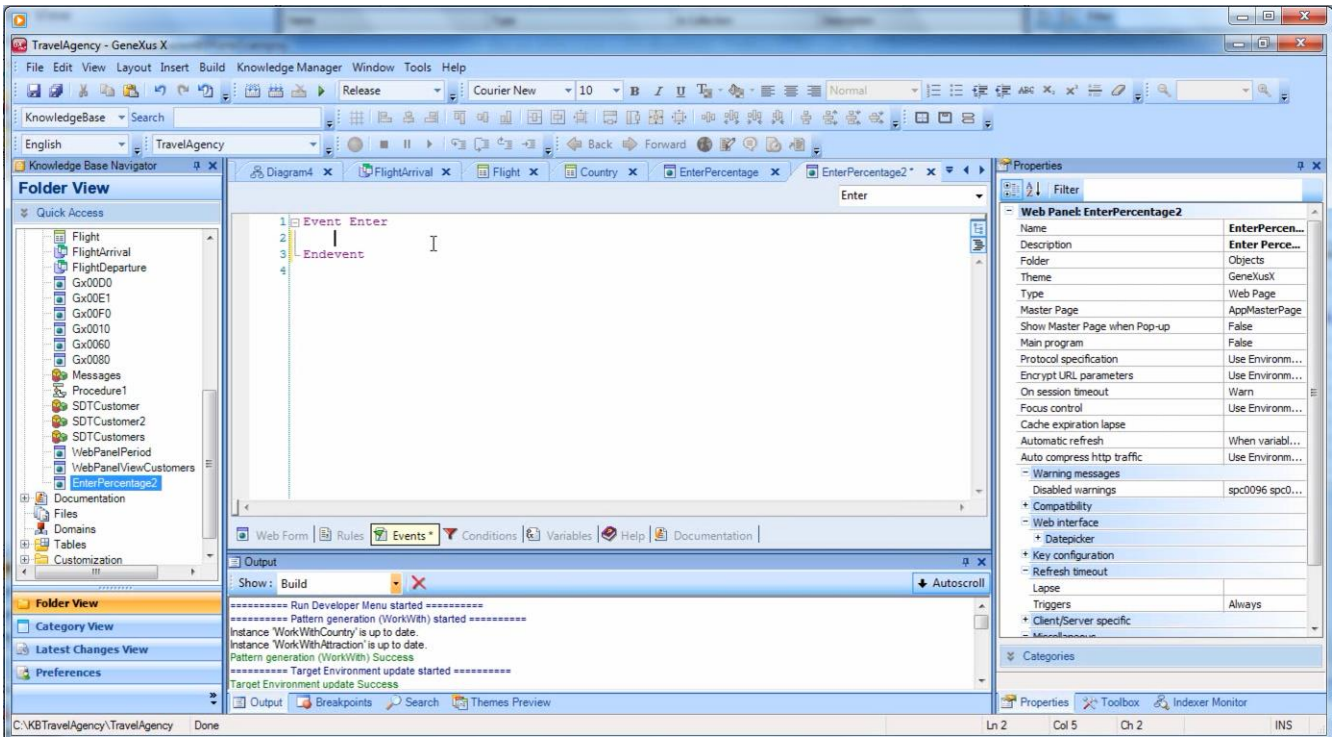


Vamos ao evento associado ao botão Confirmar



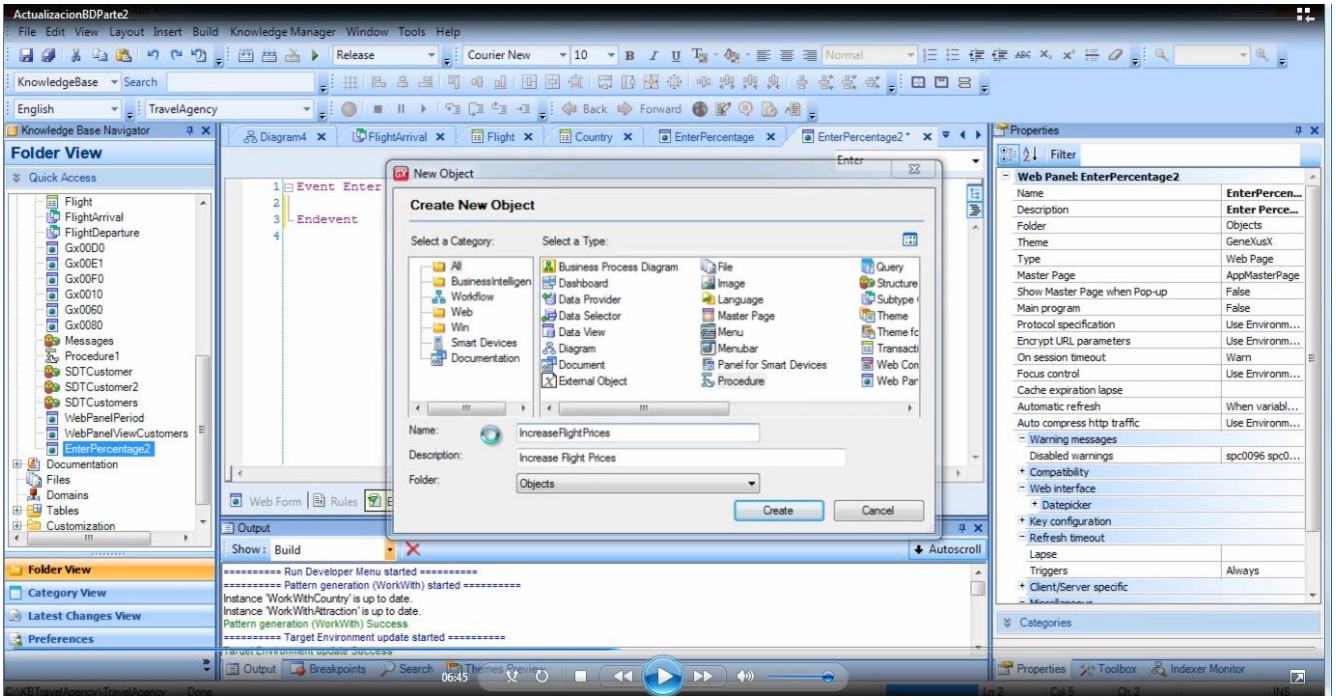
Ainda que neste evento haja um For each, a possibilidade de utilizá-lo para atualizar atributos atribuindo-lhes valores existe somente em objetos procedimentos.

Sendo assim, apaguemos este código



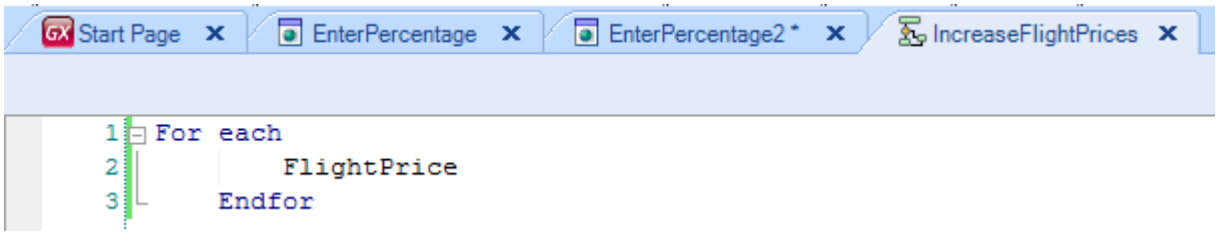
e, no evento Enter, incluiremos apenas uma chamada a um procedimento que é o que vai atualizar o banco de dados.

Vamos criar o objeto procedimento, o chamaremos de “IncreaseFlightPrices”

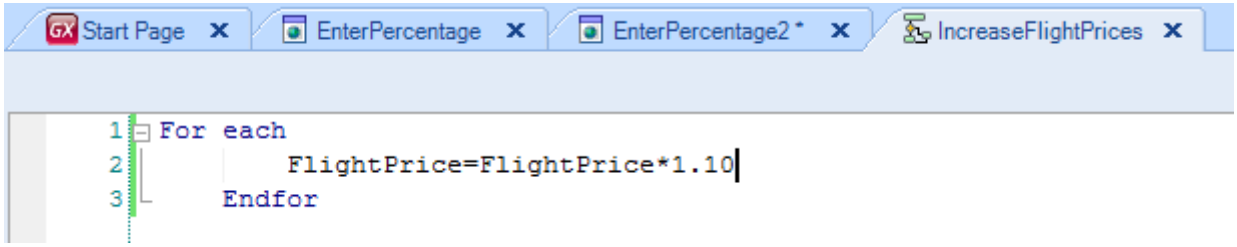


e, no source, vamos escrever o mesmo código que propusemos neste exemplo.

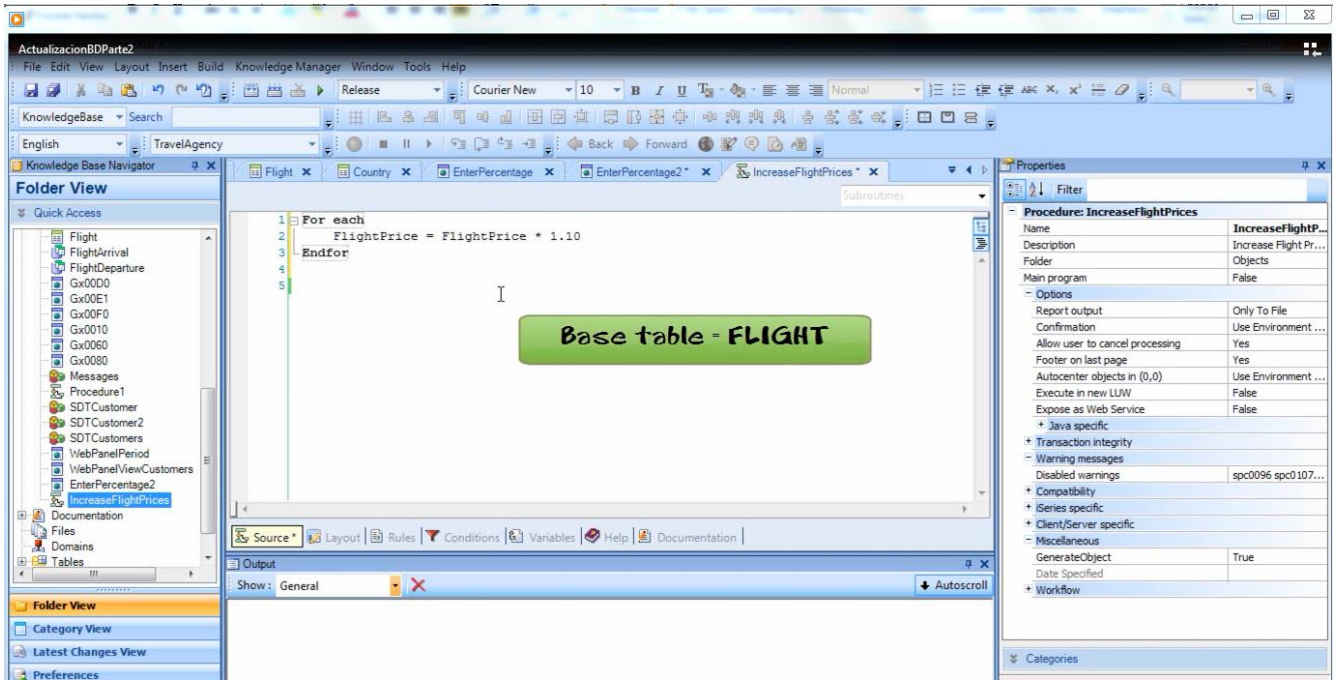
Escrevemos For each... Flight... clicamos em Control-Enter para autocompletar e escolhemos FlightPrice.



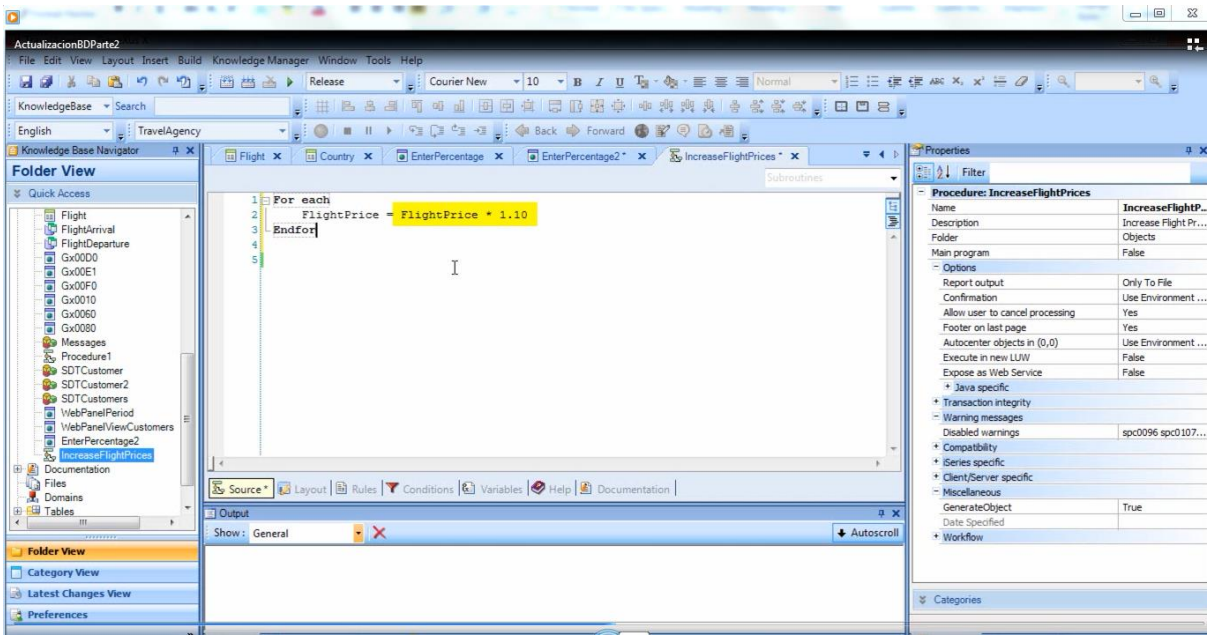
Agora, **atribuímos ao atributo FlightPrice** o valor que tinha o atributo multiplicado por 1.10.



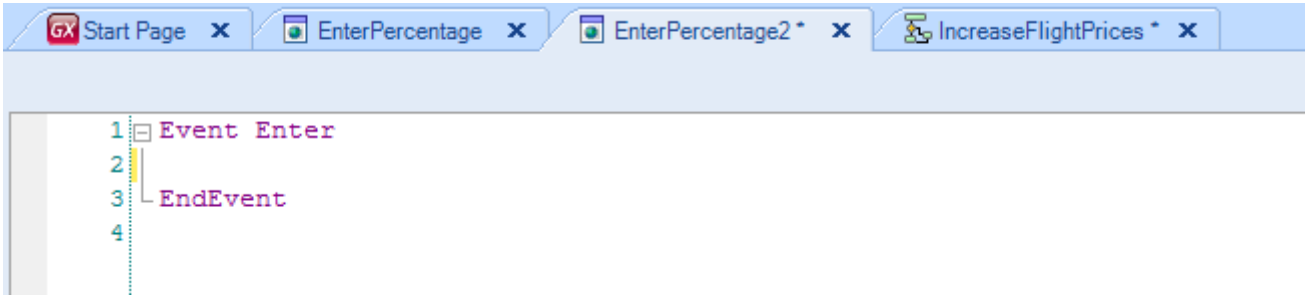
Como já explicamos, a tabela base deste For each é FLIGHT



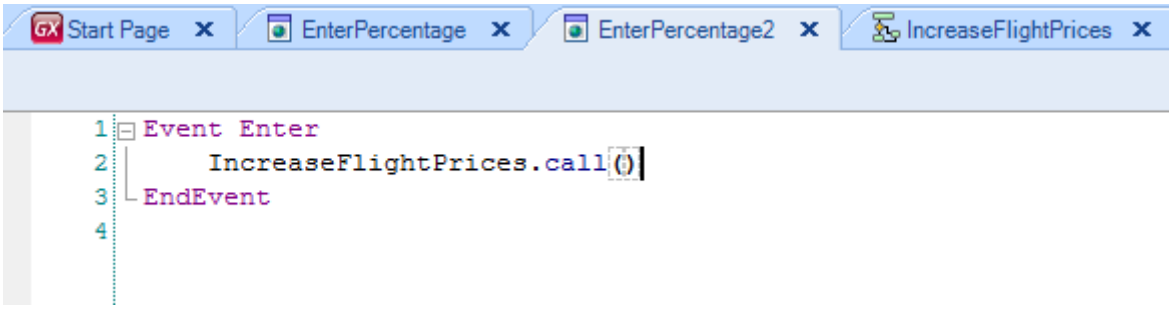
Navega-se por todos os voos e, para cada um deles, incrementamos seu preço em 10%.



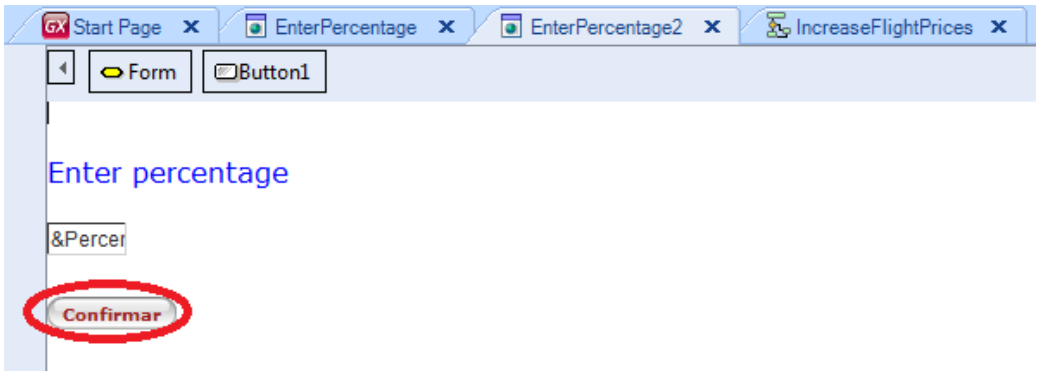
Agora, vamos chamar esse procedimento partindo do web panel.



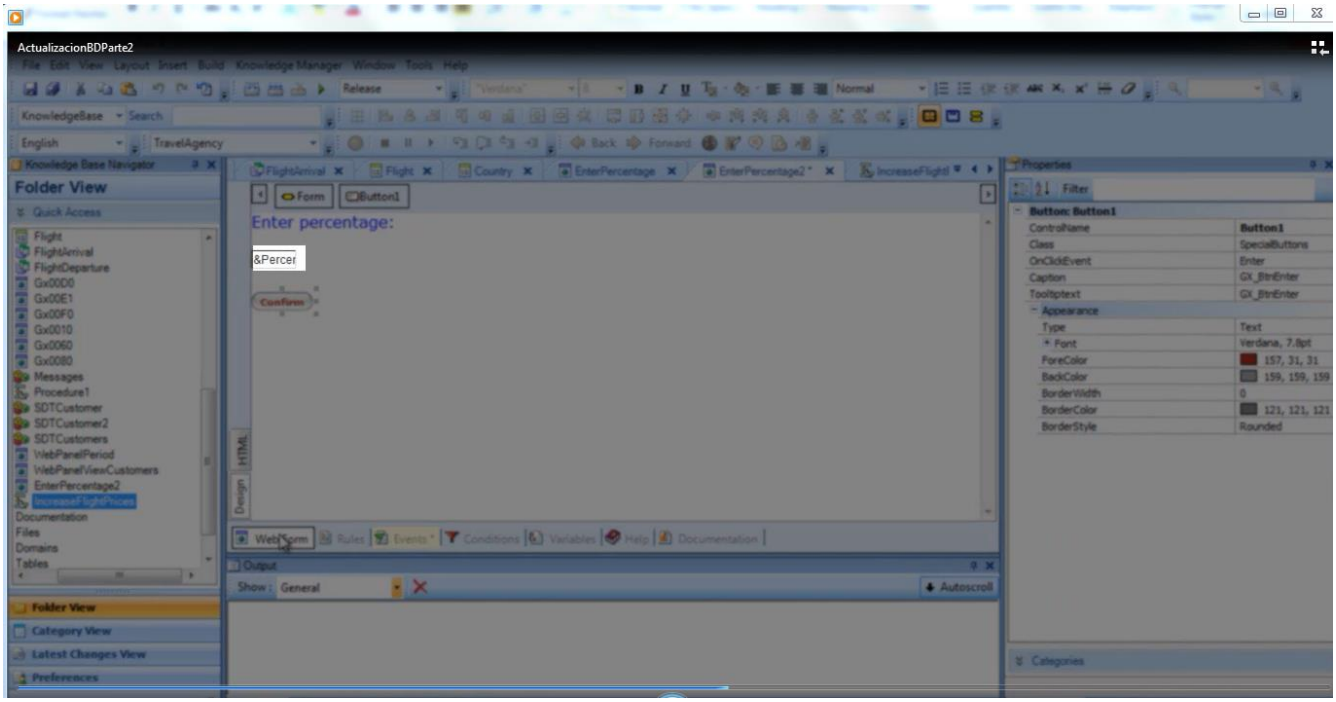
Localizamos o procedimento no Folder View e o arrastamos até o webpanel. Digitamos ponto e selecionamos o call que nos é oferecido neste menu contextual.



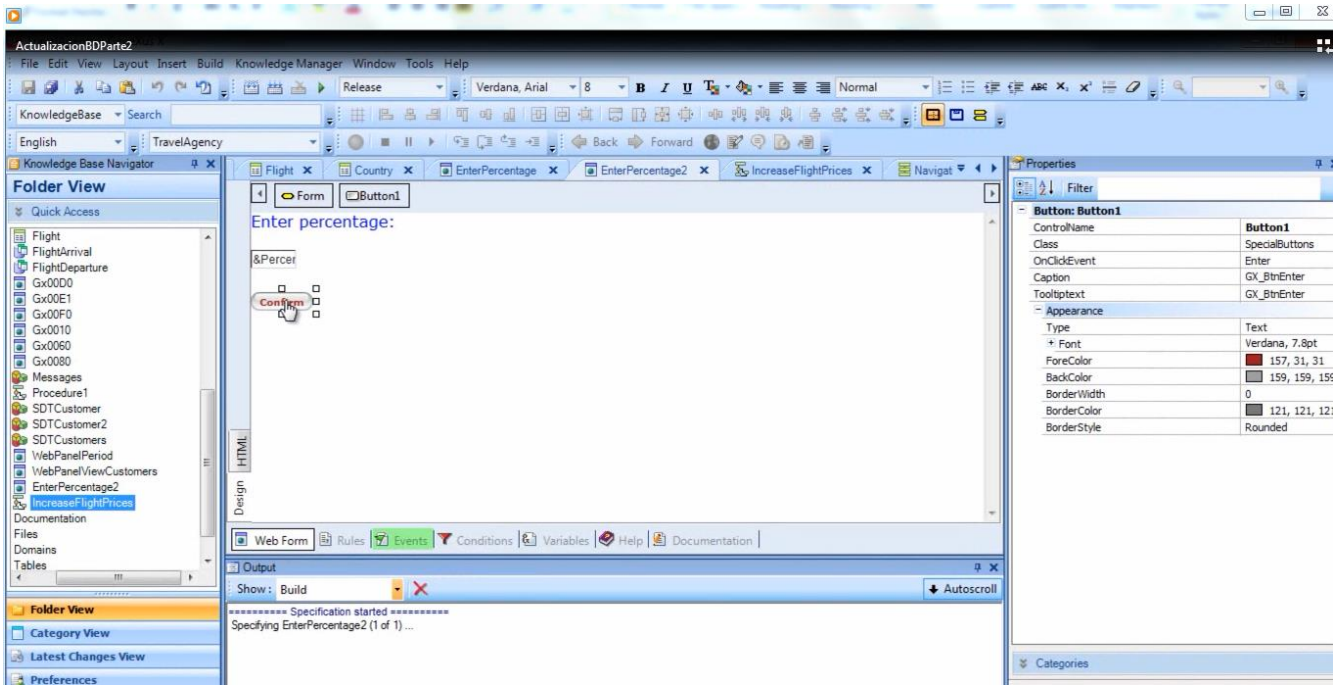
Este evento Enter será executado quando o usuário pressionar o botão associado a ele.

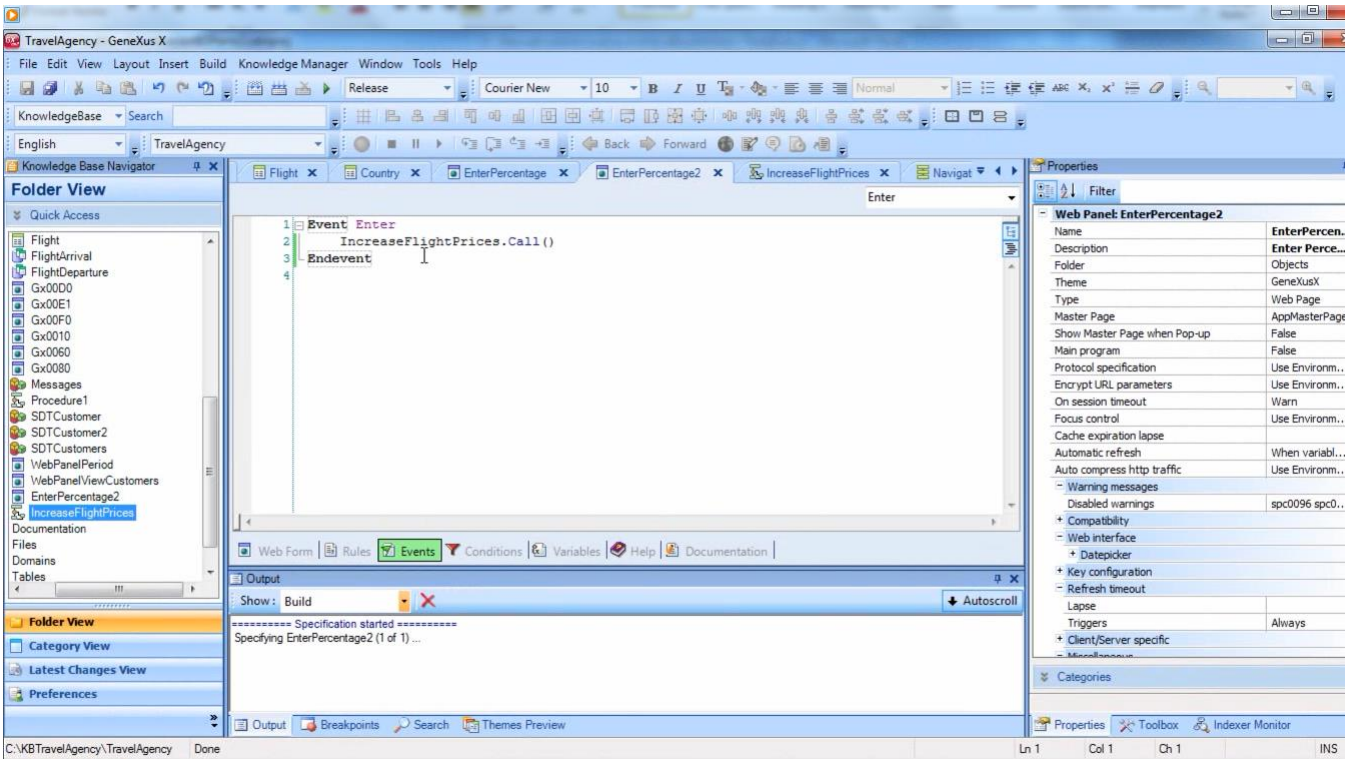


Contudo, antes de executar o que fizemos, observemos um pequeno detalhe: No web panel, o usuário poderá digitar nesta variável

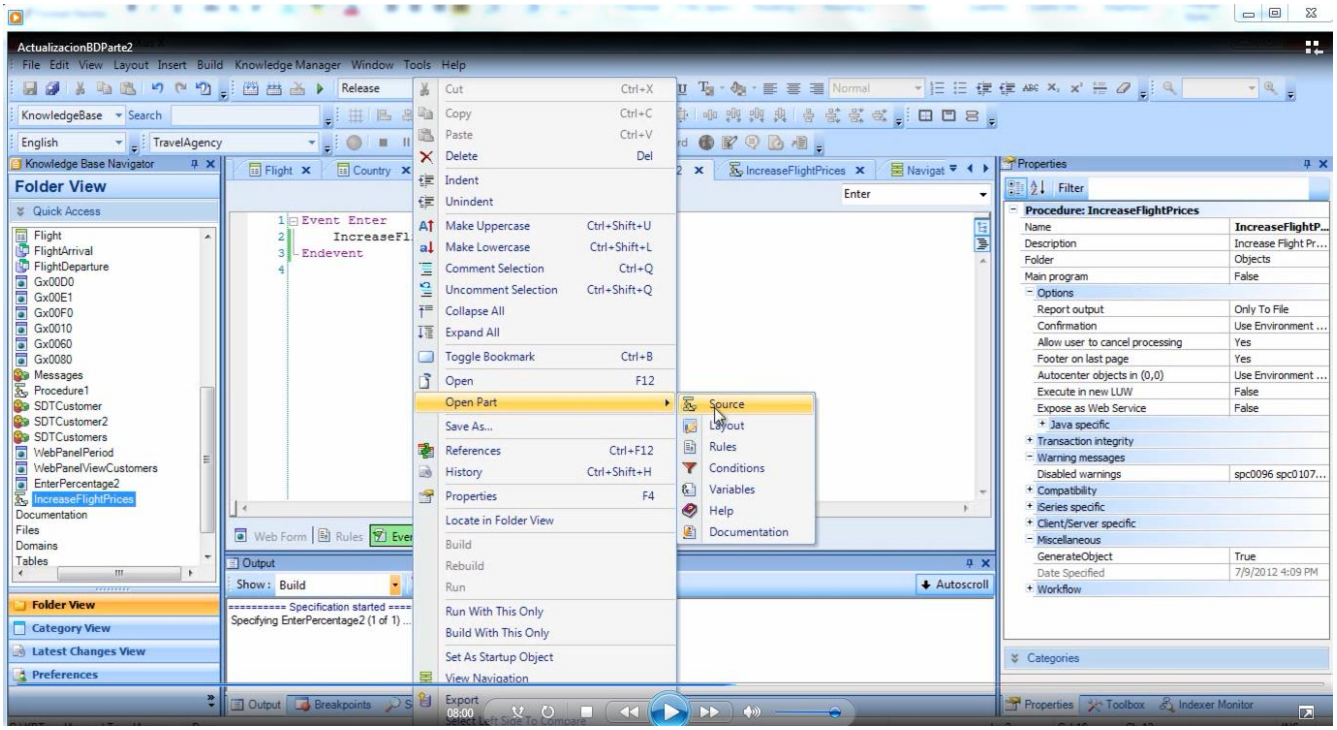


uma determinada porcentagem de aumento e, ao clicar sobre o botão,

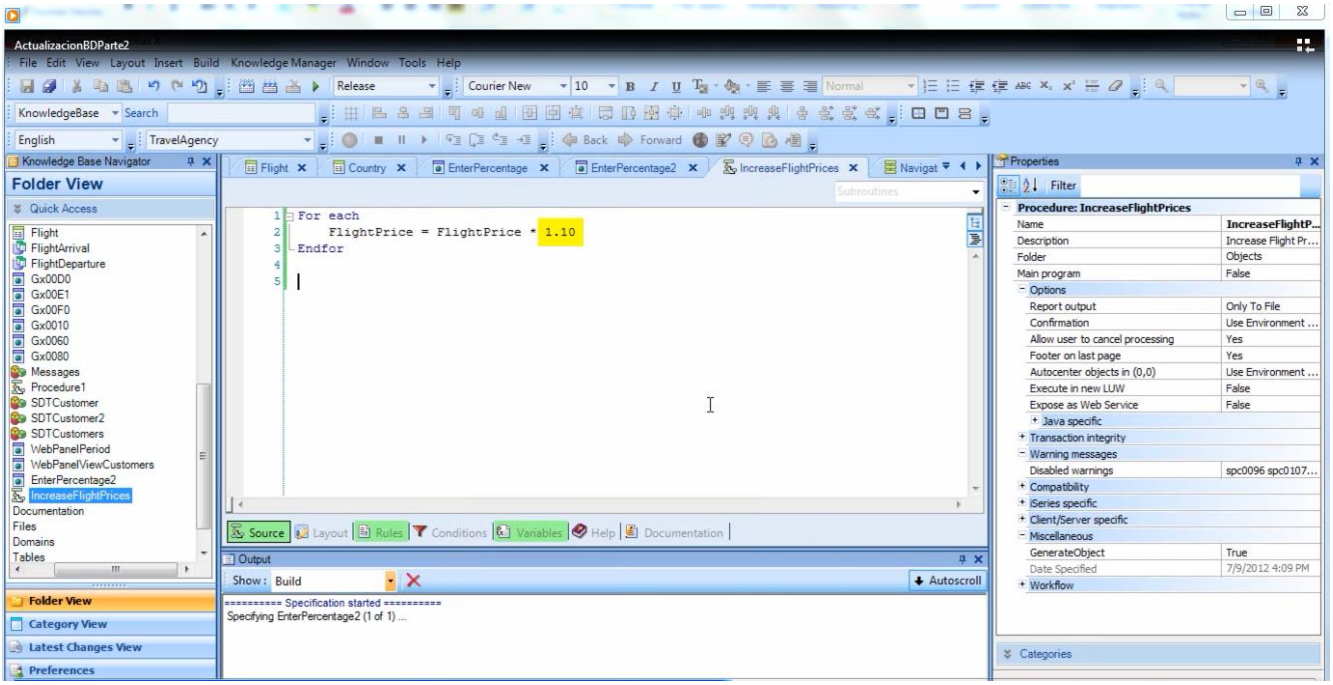




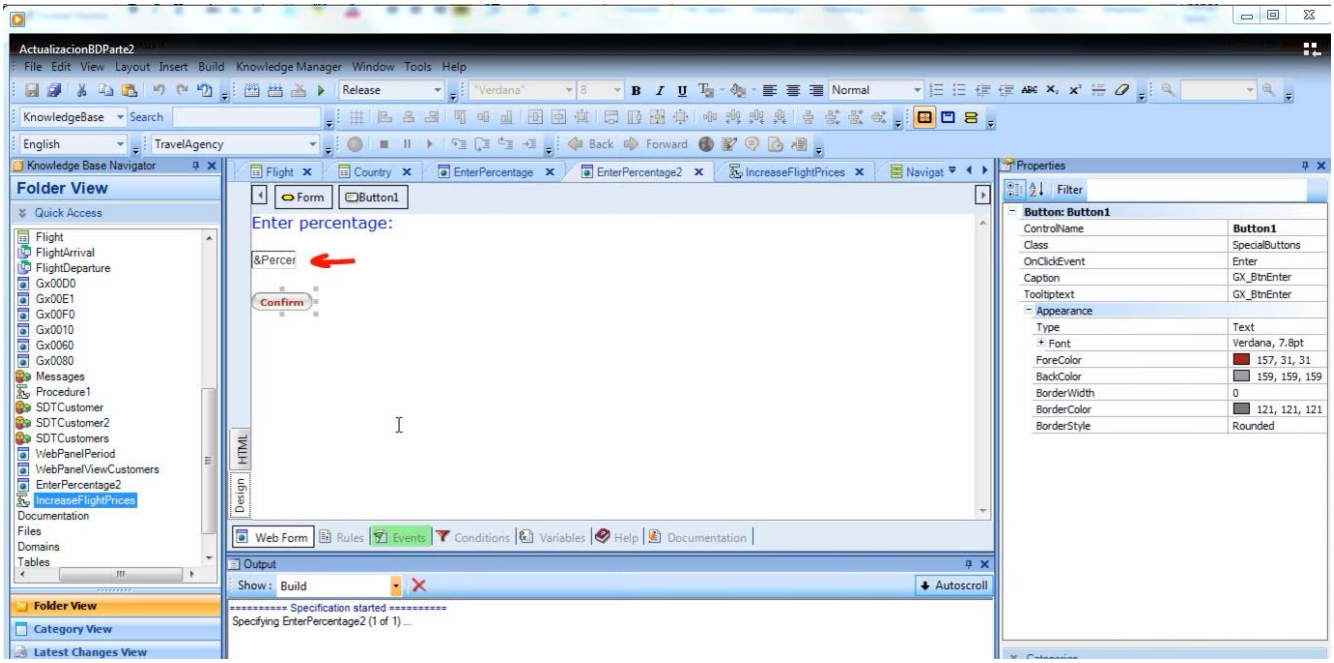
estamos chamando o procedimento



que aumenta os preços em uma porcentagem fixa de 10%.



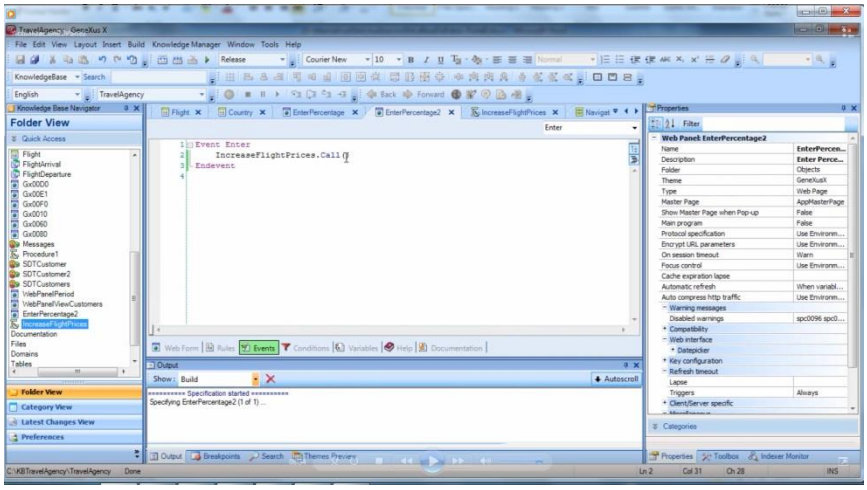
Colocamos a porcentagem de 10% em uma primeira instância como exemplo... mas agora queremos considerar o valor da porcentagem digitado pelo usuário na web panel.



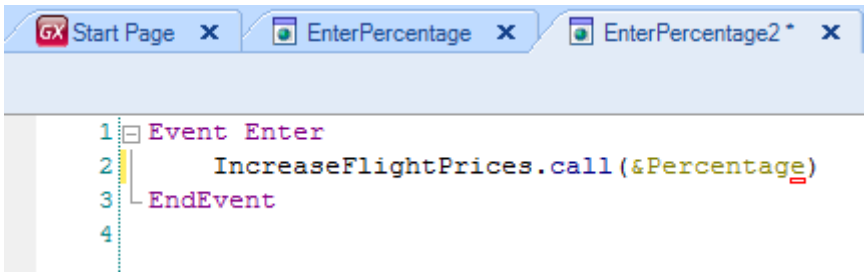
Como fazemos para que **o procedimento** conheça o valor digitado **na web panel**?

O valor da variável &Percentage é conhecido na web panel, mas não no procedimento.

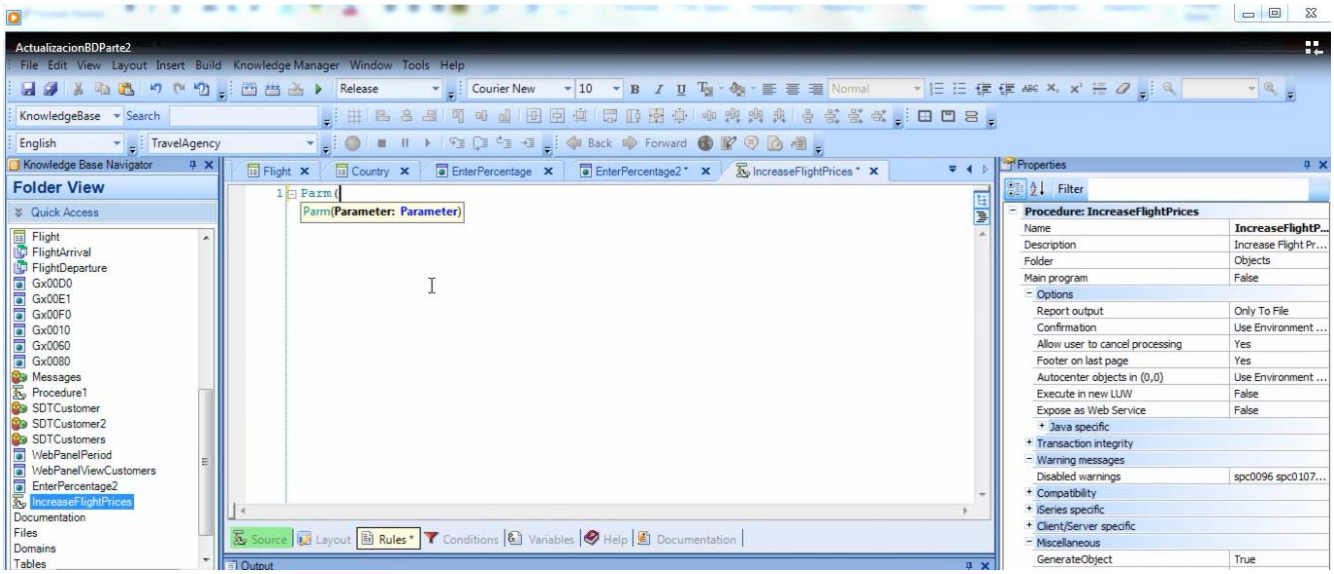
O que faremos é enviar o valor salvo na variável no momento de chamar o procedimento.



Para conseguir isso, dentro dos parênteses do call, incluímos a variável &Percentage

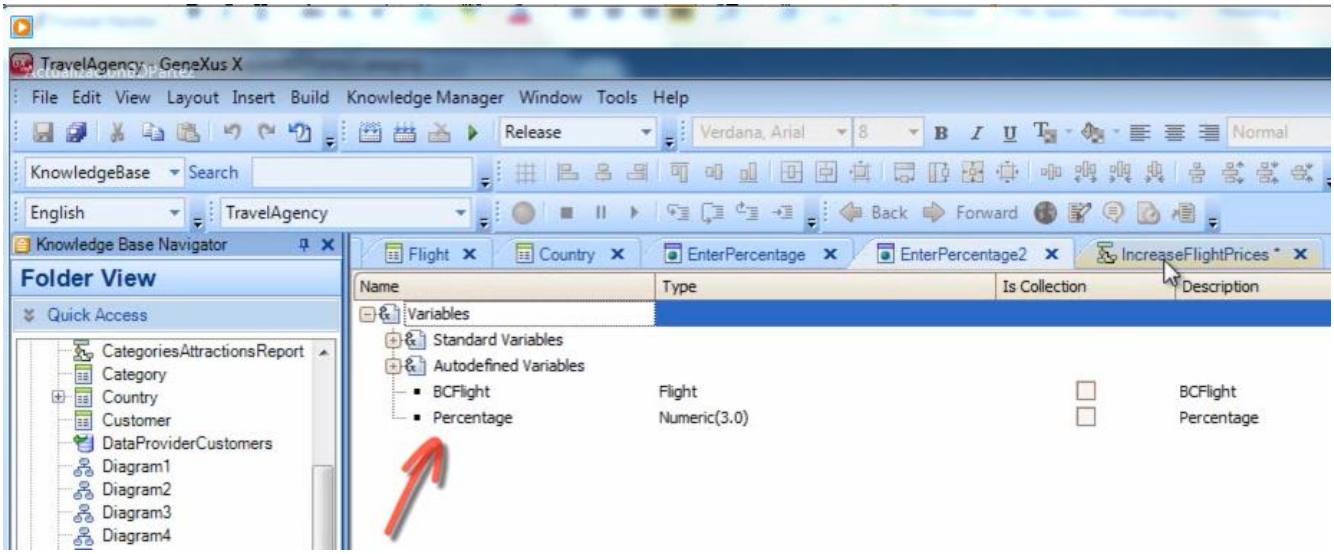


Agora, no procedimento, devemos recebê-la. Vamos ao procedimento e, na seção rules, escrevemos Parm seguido de parênteses.



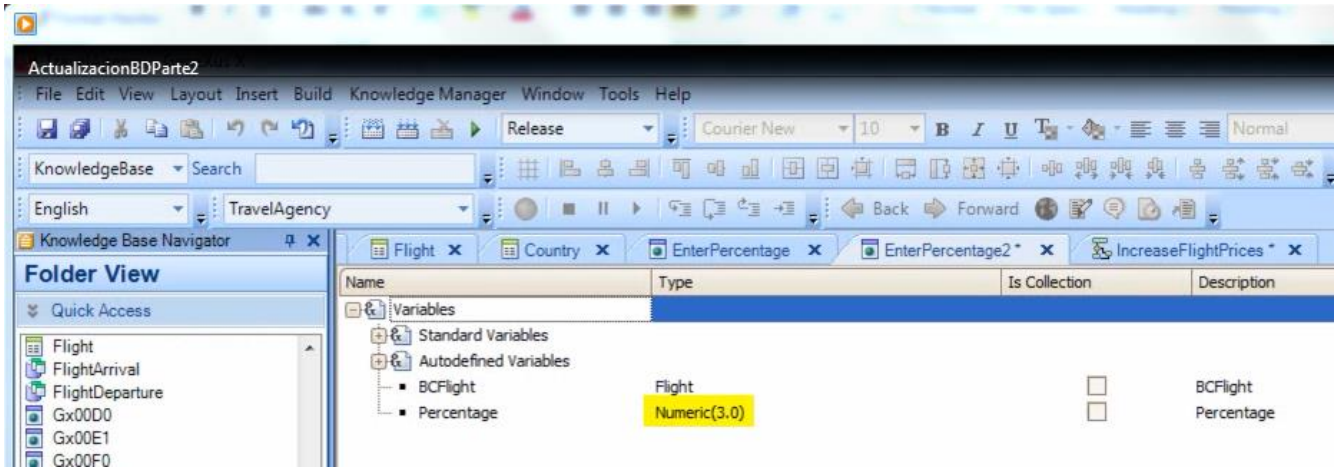
Dentro dos parênteses, temos que receber a variável, mas a variável aqui não está definida.

Podemos defini-la com o mesmo nome que está definida na web panel

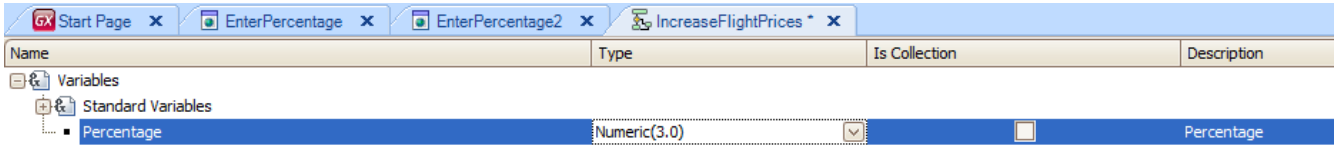


ou com outro nome. O importante é que o tipo de dados seja idêntico ao tipo de dados da variável enviada da web panel ao procedimento.

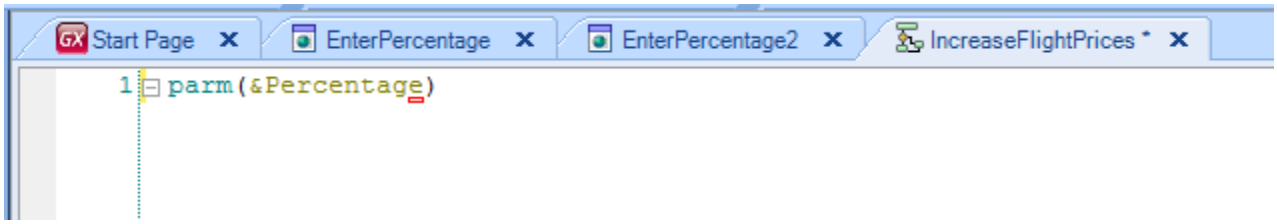
Podemos ver que o tipo de dados da variável &Percentage na web panel é: Numeric(3)



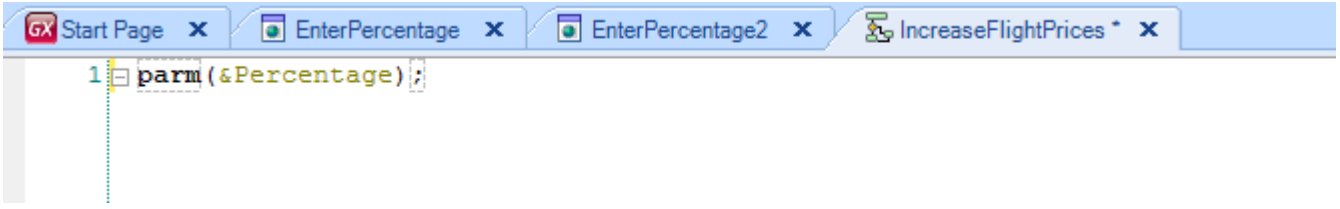
de modo que no procedimento vamos defini-la assim



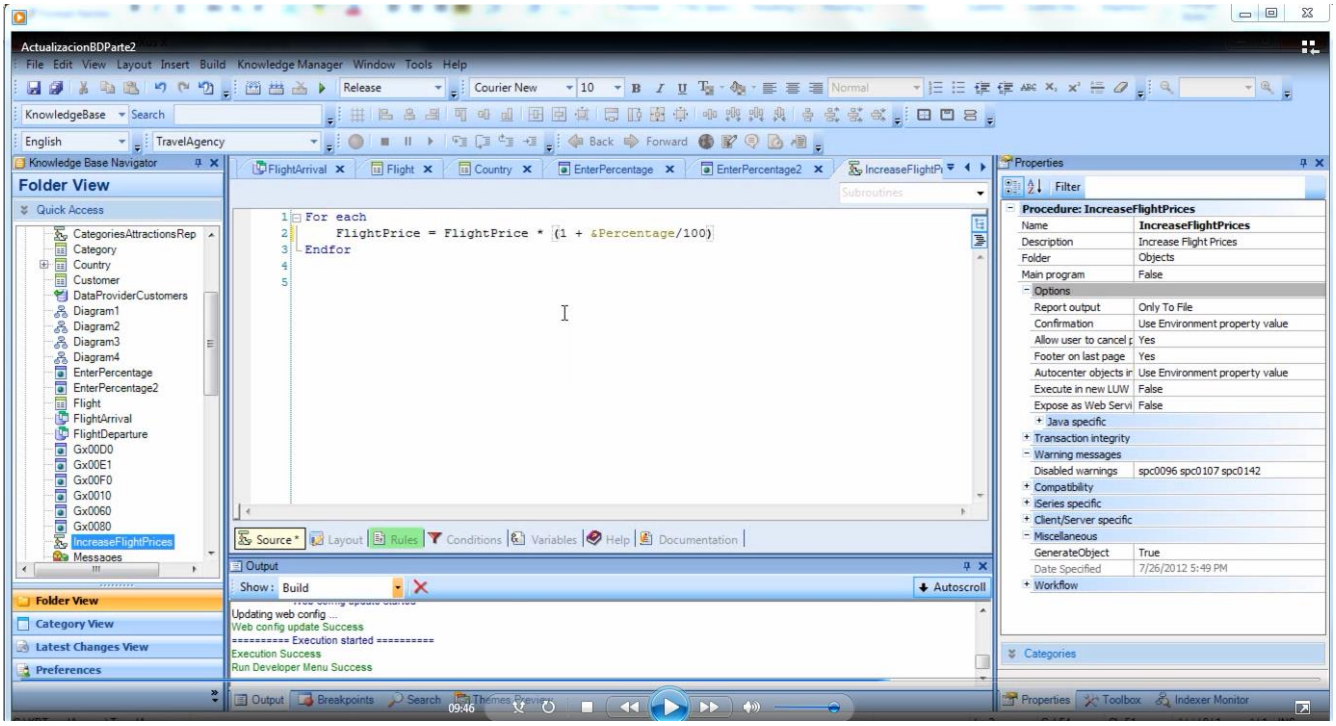
Agora, na sessão de rules, dentro dos parênteses da regra Parm, incluímos a variável que recebemos.



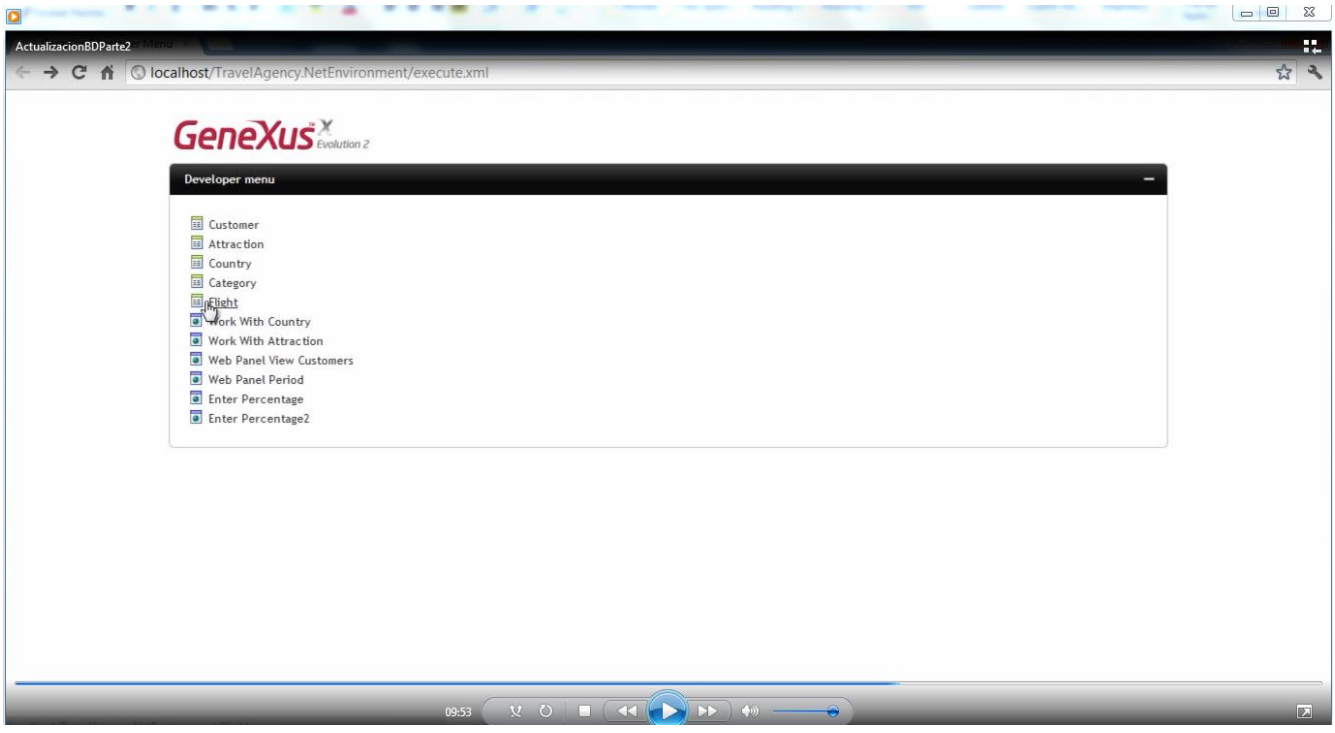
Todas as regras devem terminar com um ponto-e-vírgula em todo objeto, então o colocamos e salvamos.



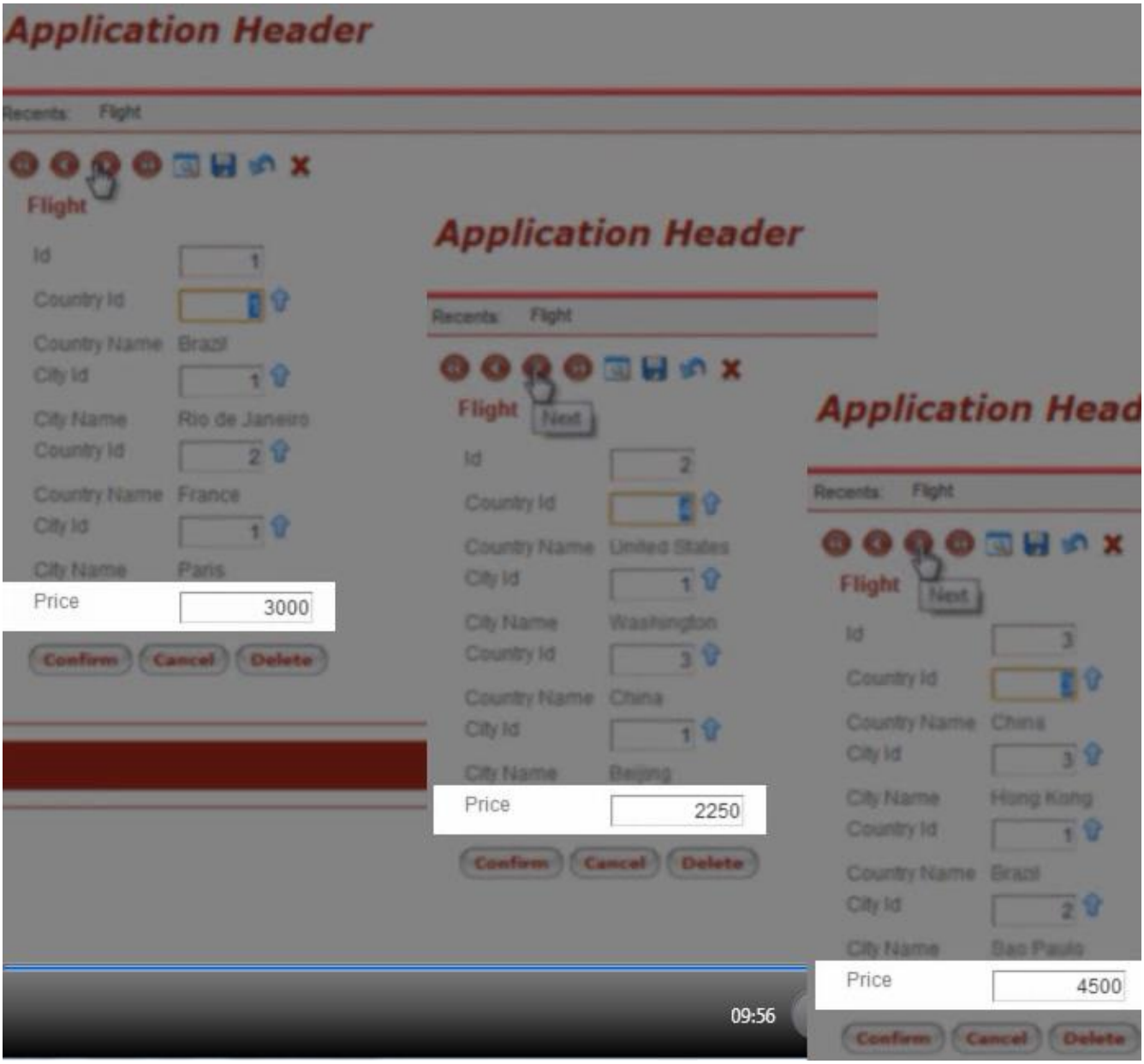
Agora vamos ao source para incluir a variável &Percentage no cálculo.



Pressionamos F5...

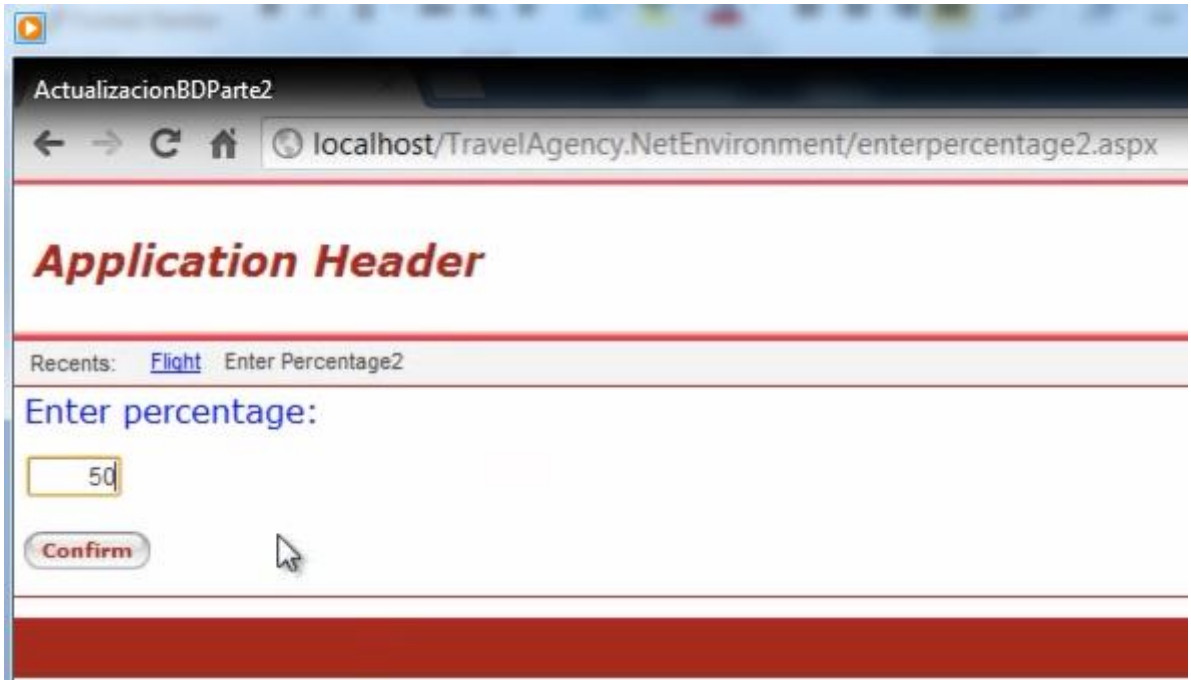


Vemos o preço atual dos voos...



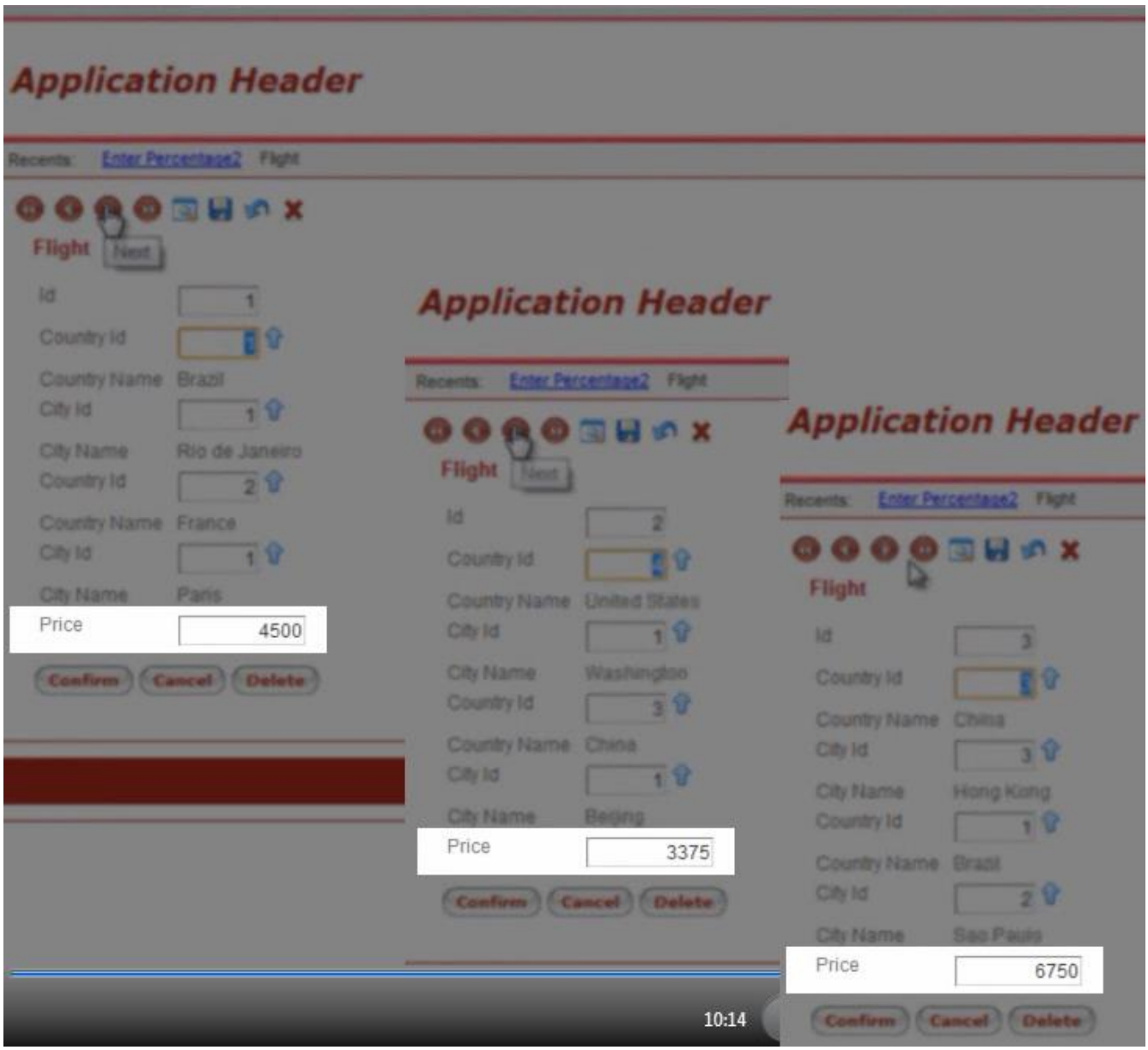
Executamos a web panel EnterPercentage2,

Inserimos 50% de aumento

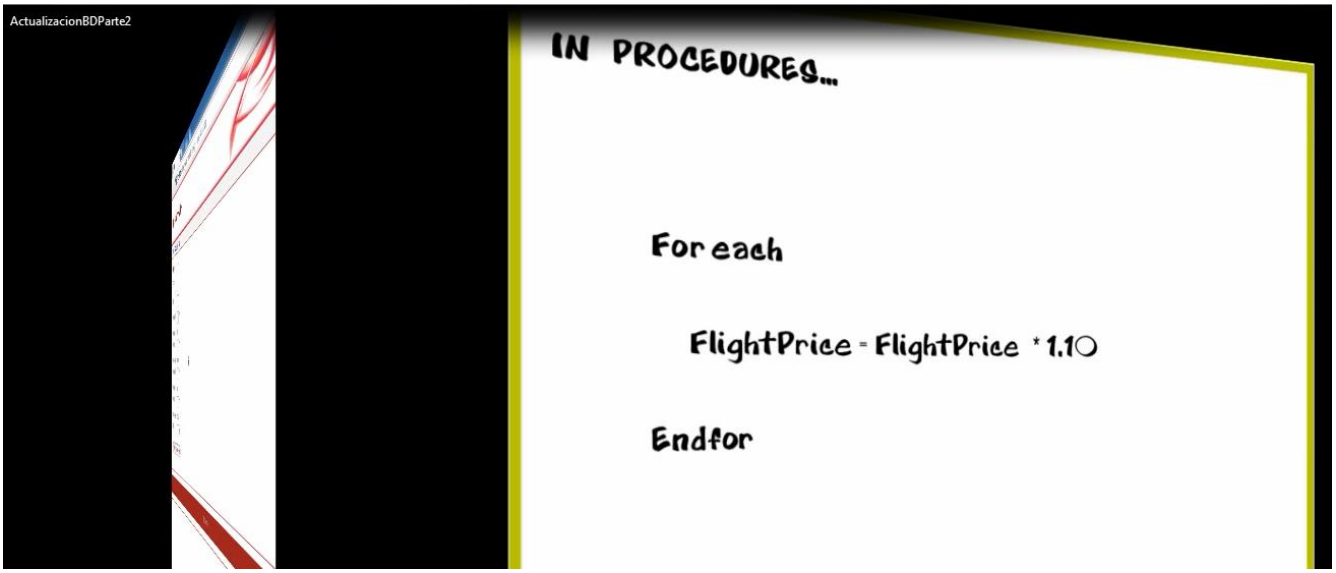


E confirmamos.

Ao vermos os preços dos voos, podemos observar que houve um aumento de 50%.

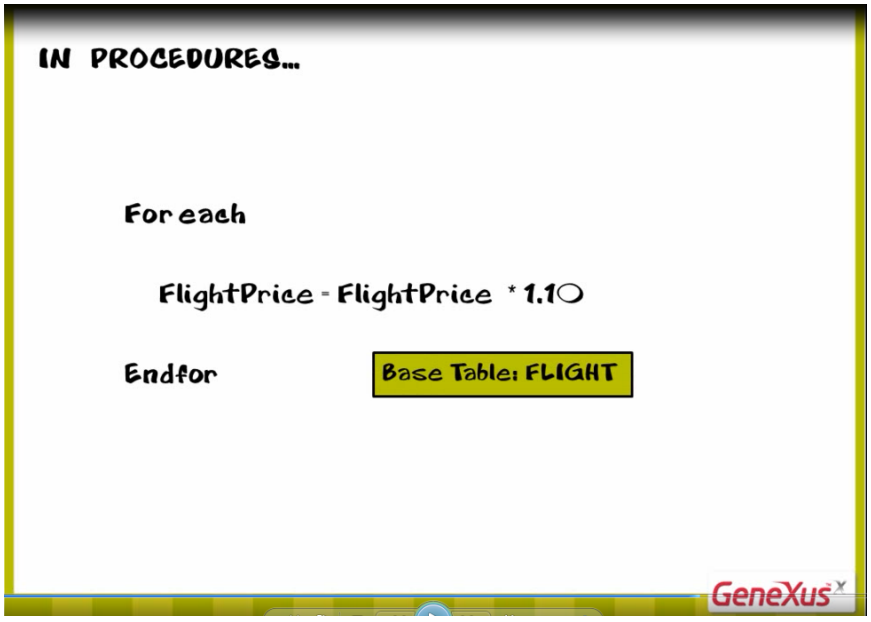

Application Header
Recents: Enter Percentage2 Flight
Flight
Next
Id 1
Country Id
Country Name Brazil
City Id 1
City Name Rio de Janeiro
Country Id 2
Country Name France
City Id 1
City Name Paris
Price 4500
Confirm Cancel Delete
Application Header
Recents: Enter Percentage2 Flight
Flight
Next
Id 2
Country Id
Country Name United States
City Id 1
City Name Washington
Country Id 3
Country Name China
City Id 1
City Name Beijing
Price 3375
Confirm Cancel Delete
Application Header
Recents: Enter Percentage2 Flight
Flight
Id 3
Country Id
Country Name China
City Id 3
City Name Hong Kong
Country Id 1
Country Name Brazil
City Id 2
City Name Sao Paulo
Price 6750
Confirm Cancel Delete
10:14

É importante que se saiba

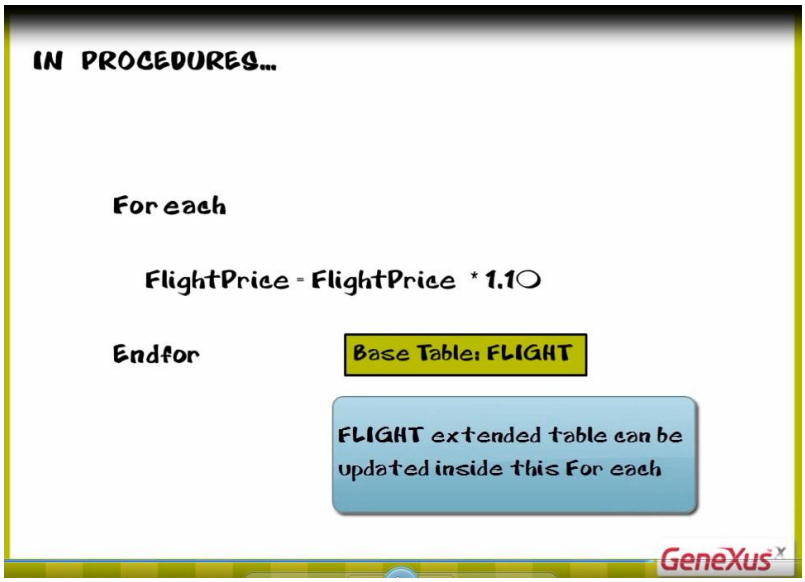


que em um mesmo For each podemos atualizar **várias tabelas físicas.**

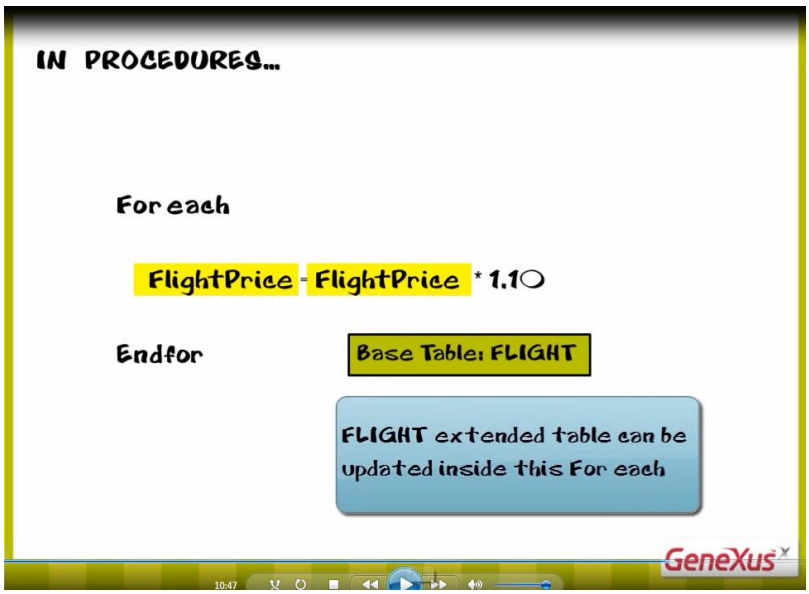
Concretamente, um For each tem sempre uma tabela base pela qual navega e que pode modificar,



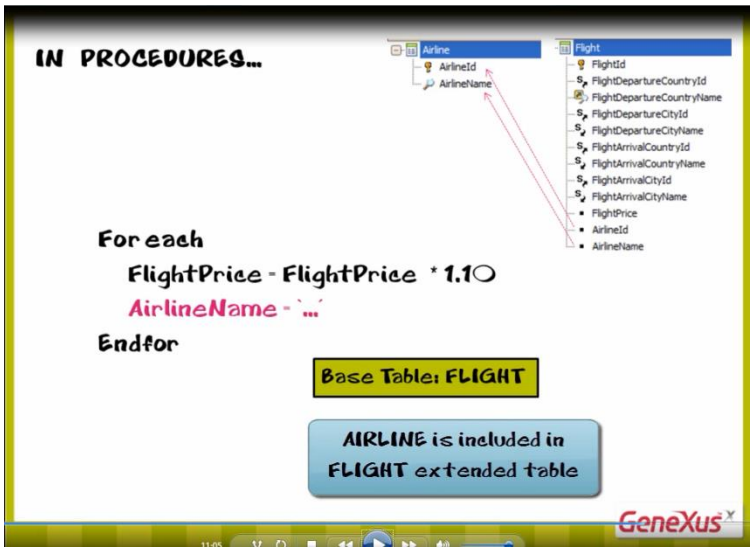
mas também pode modificar toda a tabela estendida desta tabela base.



Neste exemplo, a tabela base do For each é Flight, já que dentro do For each está presente apenas o atributo FlightPrice



Como sabemos cada voo tem um país de origem, um país de destino, uma cidade de origem e uma cidade de destino. Neste For each, portanto, poderíamos modificar os dados de tais países e cidades, se assim fosse necessário. Ou, por exemplo, se cada voo tivesse uma empresa aérea,



poderíamos modificar dentro deste For each, para cada voo, os dados da empresa associada ao voo.

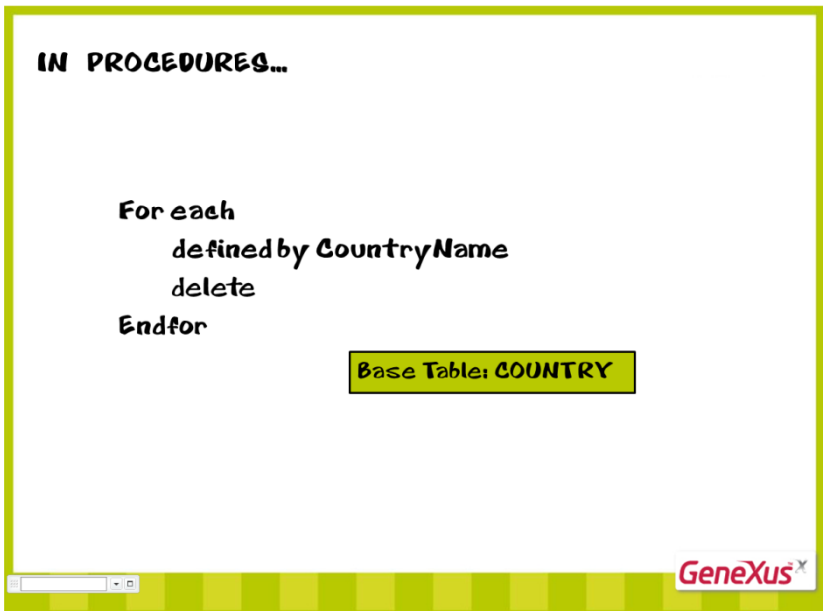
Já vimos como os procedimentos permitem-nos incluir e atualizar registros no banco de dados. Vejamos agora como excluir registros.

Para excluir registros, contamos com o comando Delete

que é usado dentro de um comando For each.

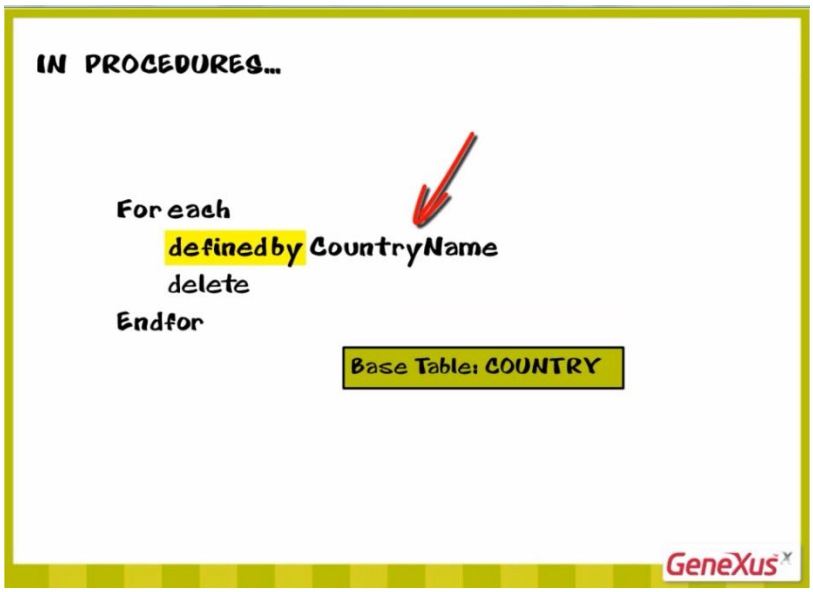
Basicamente, é preciso navegar pela tabela na qual estamos querendo eliminar um ou vários registros e incluir dentro do For each o comando delete.

No exemplo, vemos que estamos navegando pela tabela Country, filtrando por um determinado país e, portanto, eliminando pontualmente um registro com o comando Delete. Contudo, também poderíamos ter excluído todos os países inseridos na tabela Country.



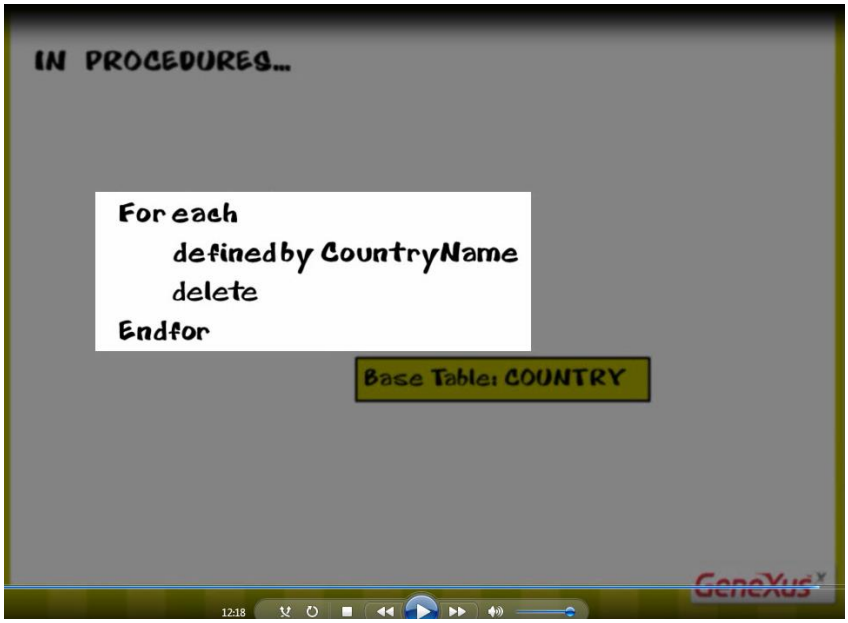
Visto que não necessitávamos filtrar determinados registros, nem ordenar, nem imprimir, o For each nos oferece a cláusula **defined by**

para poder referenciar um atributo que ajude a determinar a tabela base



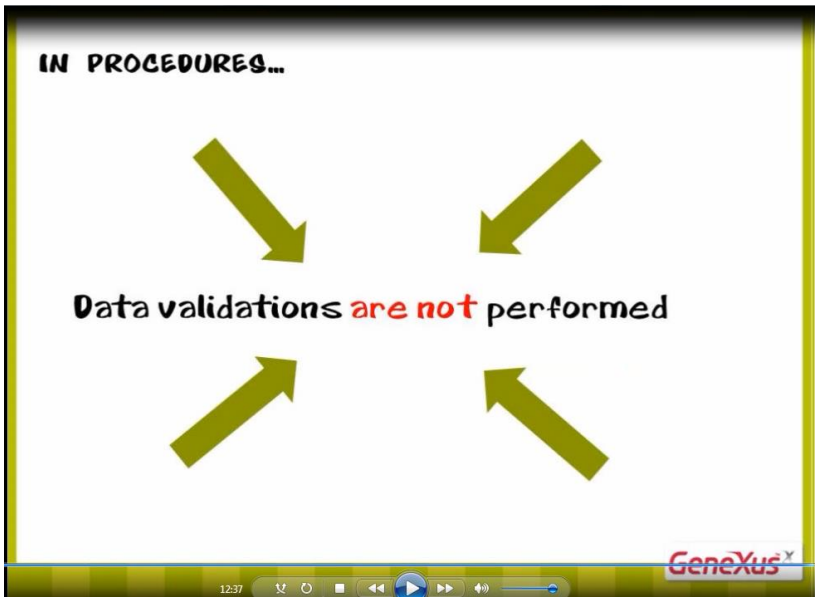
Como já explicamos, os procedimentos não levam em conta os dados relacionados em outras tabelas.

Assim, este For each que exclui todos os países

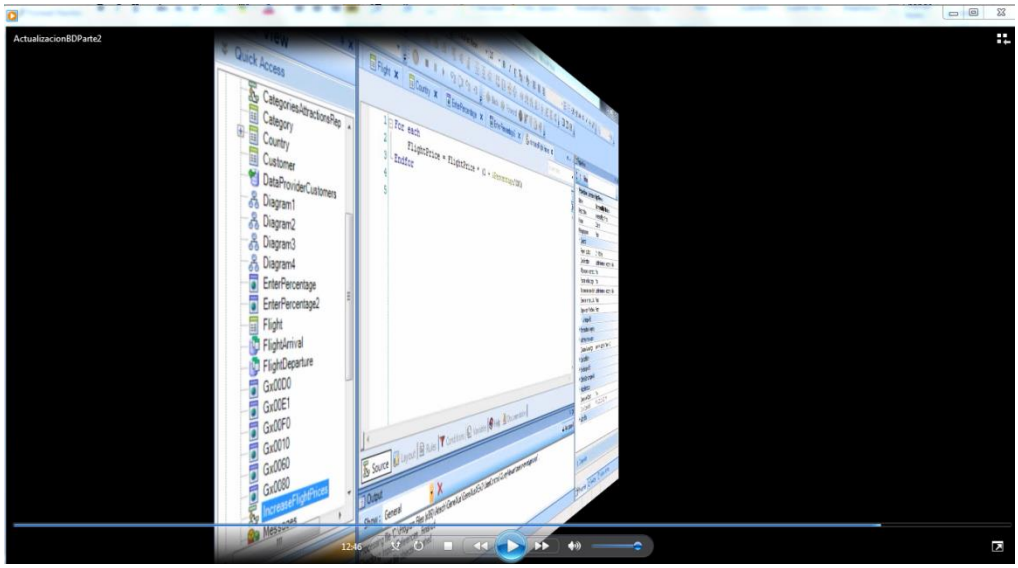


ao ser executado, poderia deixar atrações turísticas armazenadas referenciando países que tenham sido apagados.

Como o banco de dados controla a consistência dos dados inter-relacionados, será negada a operação e o programa cancelará sua execução.

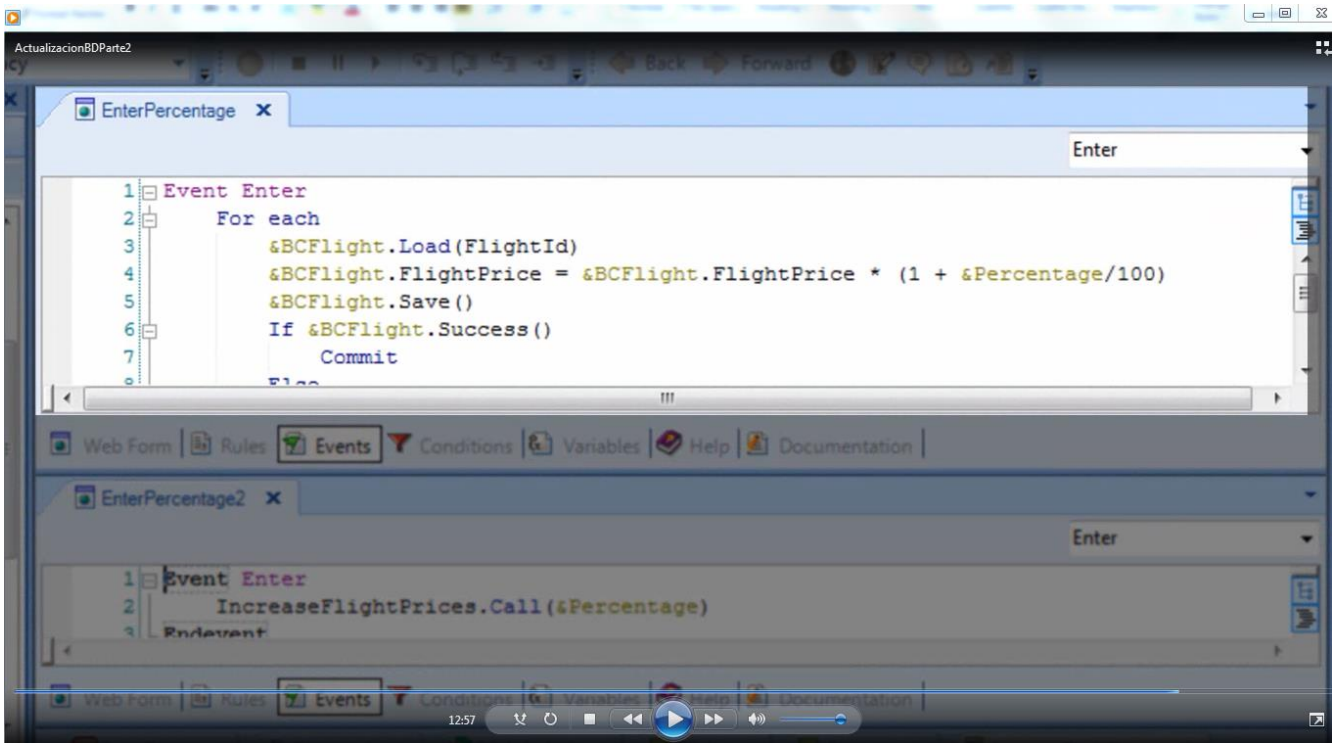


É, portanto, nossa responsabilidade ao usar procedimentos, apagar, atribuir ou incluir dados que sejam coerentes com o resto da informação armazenada.



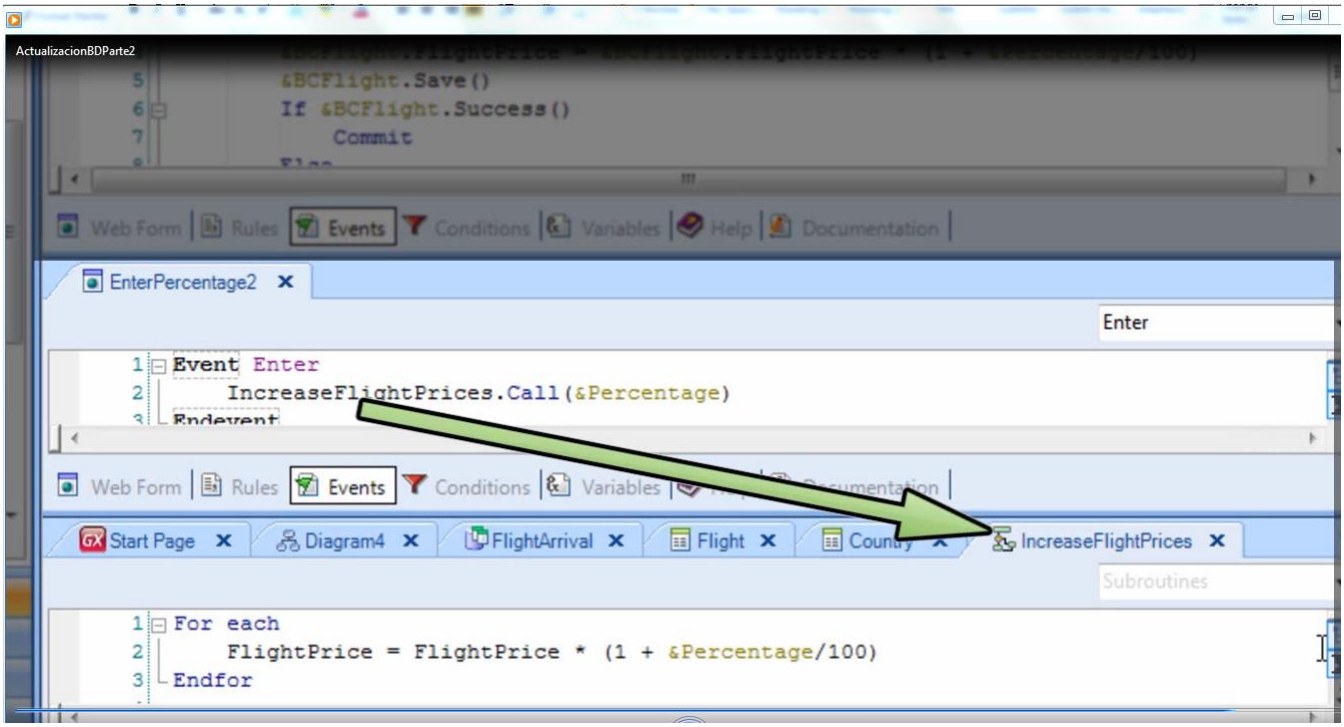
Para finalizar, comparemos as duas alternativas que utilizamos para atualizar o banco de dados.

Na primeira alternativa que implementamos,

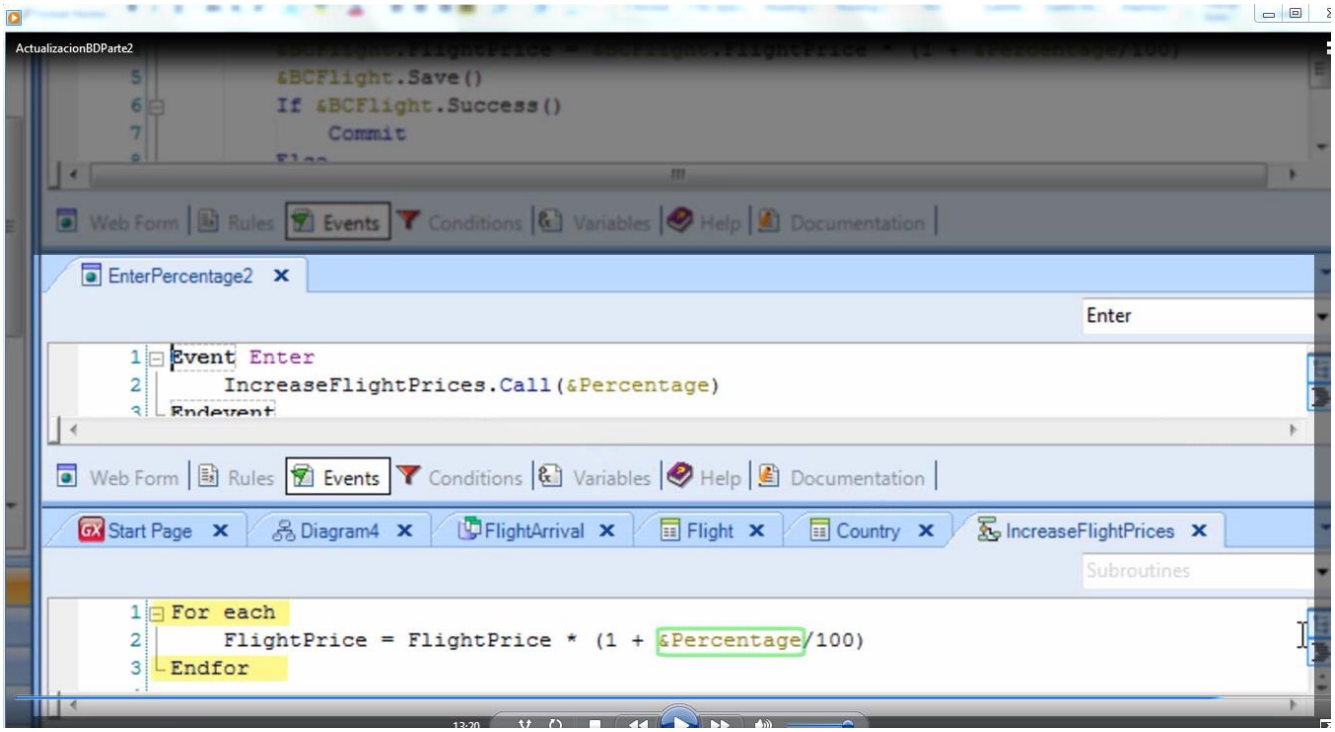


no evento enter do web panel EnterPercentage, escrevemos o For each e atualizamos o banco de dados empregando a transação Flight como Business Component.

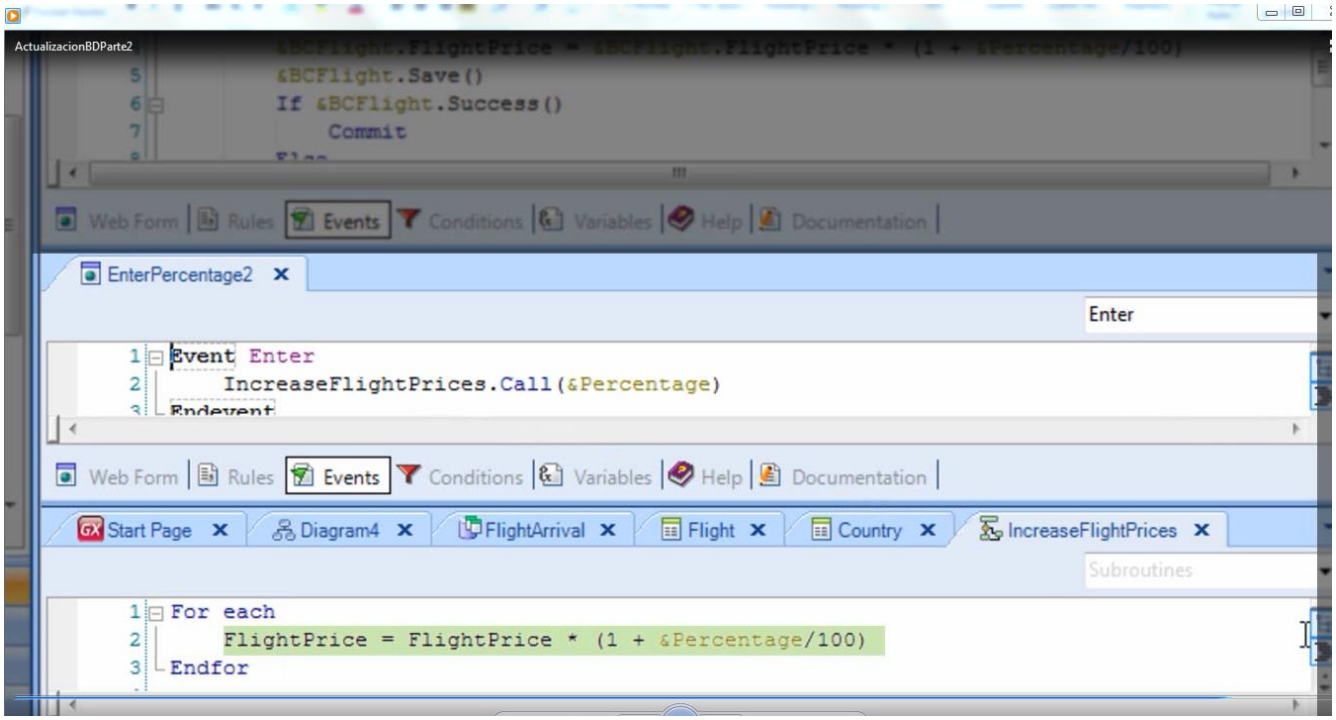
Na segunda solução, o evento do web panel EnterPercentage2, contém apenas uma chamada a um procedimento,



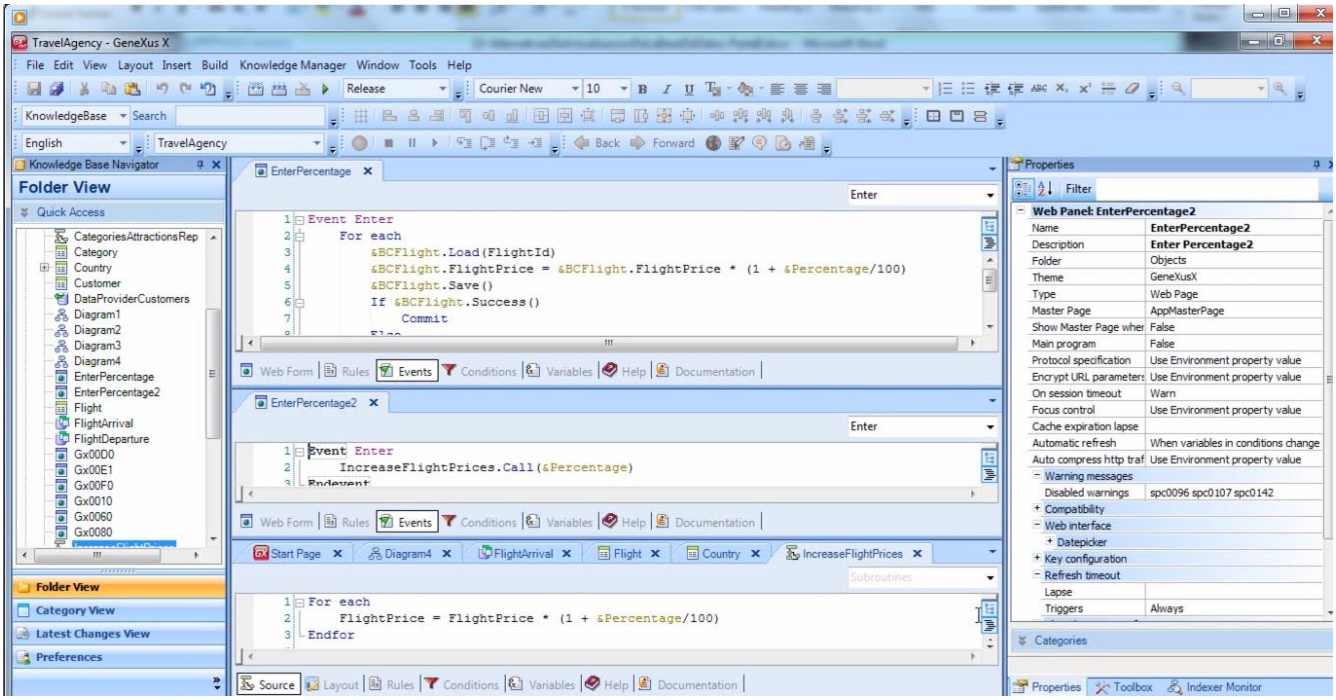
o procedimento recebe o valor da porcentagem, navega com For each



todos os voos e, para cada um deles, calcula e lhe atribui o novo preço.



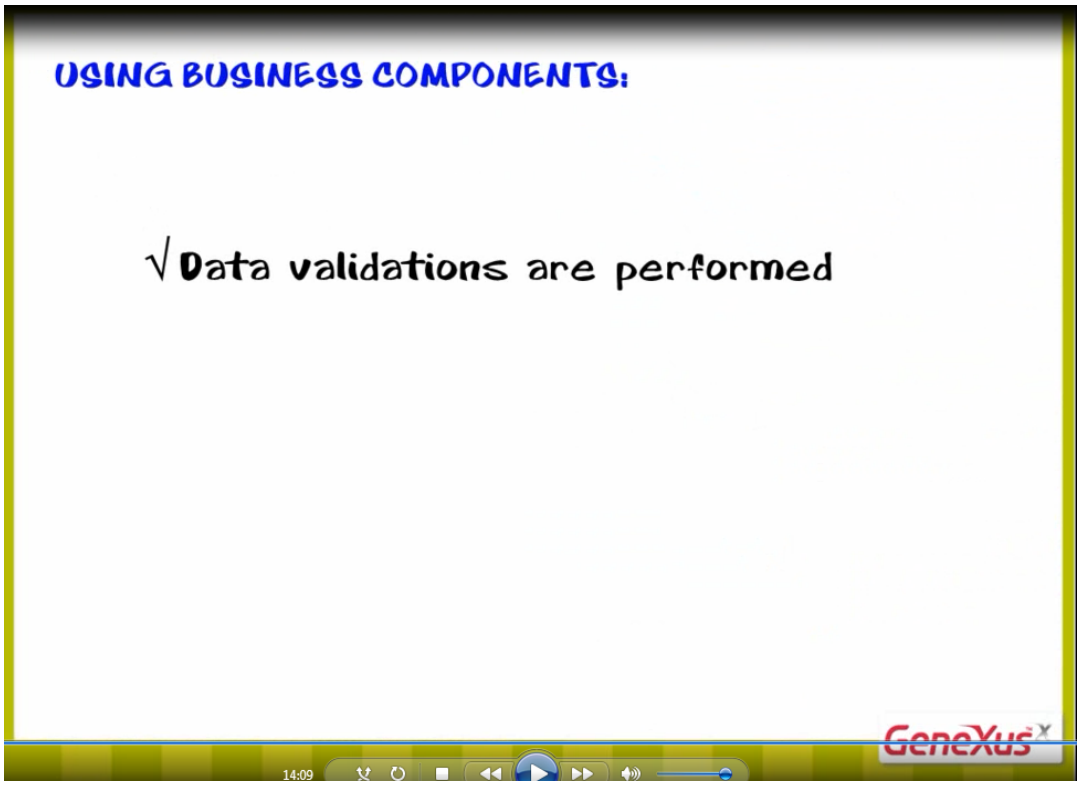
Quais são as diferenças, vantagens e desvantagens em resolver de uma maneira ou de outra?



Algo que vimos e que vale a pena relembrar é: embora o comando For each possa ser usado em um web panel, não é possível utilizá-lo para modificar o banco de dados diretamente atribuindo valores aos atributos, e que isso só pode ser realizado partindo-se de um objeto procedimento. Tampouco é possível codificar um comando New em um web panel, nem incluir um comando delete dentro do For each.
Isso é válido apenas em procedimentos.

Por outro lado é possível, em qualquer objeto, modificar o banco de dados com business components.

Além disso, quando empregamos business components



valida-se a consistência dos dados que se tenta atualizar no banco de dados

e executam-se as regras definidas na transação que se está executando como business components.

As regras que geram mensagens como msg e erro também são disparadas, e as mensagens correspondentes ficam guardadas em um grupo ao qual se pode recorrer e imprimir.

Em um procedimento, nenhum dessas coisas é possível.

Desta maneira, vimos diferentes alternativas para atualizar o banco de dados em um aplicativo GeneXus.