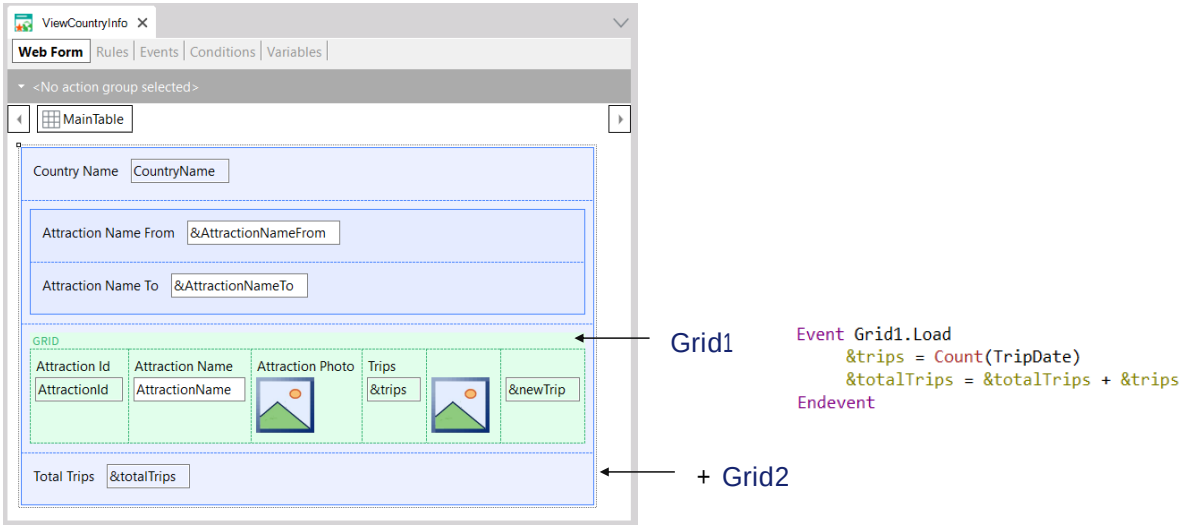
Telas web com foco em Back-Office

Objeto Web Panel. Múltiplos grids

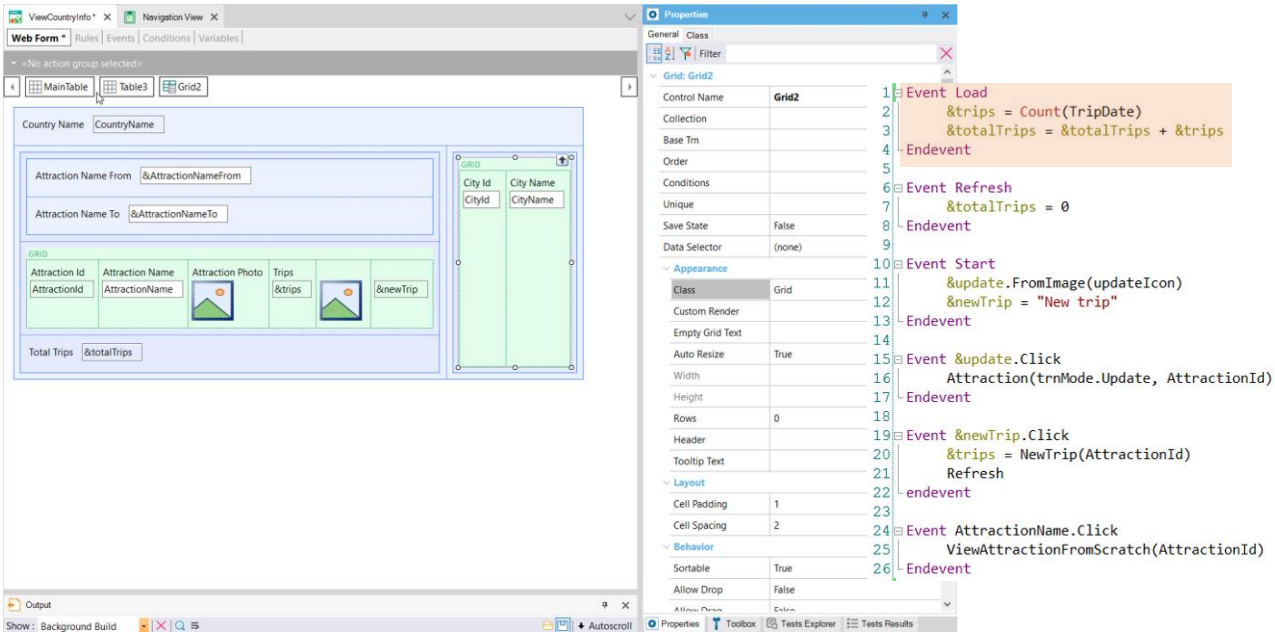
GeneXus

Mais de um grid



Dissemos algumas vezes que talvez tivesse sido melhor utilizar o evento Load do grid e não o genérico, que só serve no caso de um web panel sem grid ou com um grid. Usando o Load do grid, nos antecipamos ao futuro, para a possibilidade de inserir mais um grid.

Novo grid



Suponhamos que no web panel que mostra a informação de um país (seu nome e suas atrações turísticas), queremos adicionar também um grid com suas cidades. Antes de fazer isso, vejamos que sua lista de navegação indica a carga do -neste momento- único grid.
Antes de adicionar o grid para as cidades, inserimos dentro de uma tabela tudo o que correspondia às atrações do país, para que fique toda essa informação junta.

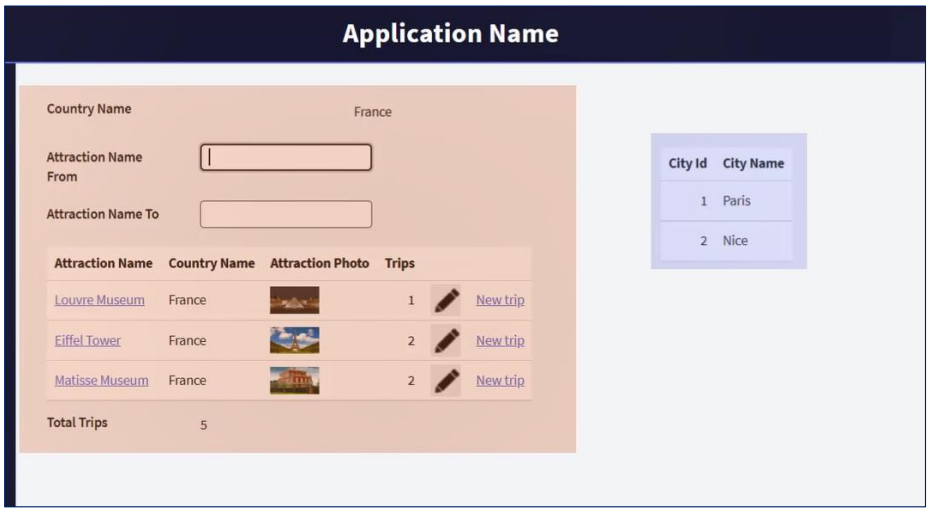
Depois, inserimos outra tabela para a informação das cidades. Ali dentro inserimos o novo grid, composto pelos atributos CityId e CityName.
Se observamos suas propriedades, vemos que o nomeou por padrão como Grid2.

Cada grid poderá ter ou não tabela base. Neste caso, ambos os grids têm atributos, assim ambos terão tabela base. Como se sabe a qual de ambos se aplica o código do evento Load genérico? De fato, se gravamos, vemos que a lista de navegação mostra um erro advertindo-o.

Está mostrando as navegações que deverá realizar para carregar cada grid, e até entendeu que a fórmula para calcular as trips deve pertencer à carga do Grid1, mas nos pede que determinemos isto. E é o que faremos.

Agora, ao gravar o objeto, a lista de navegação já não mostra o erro.

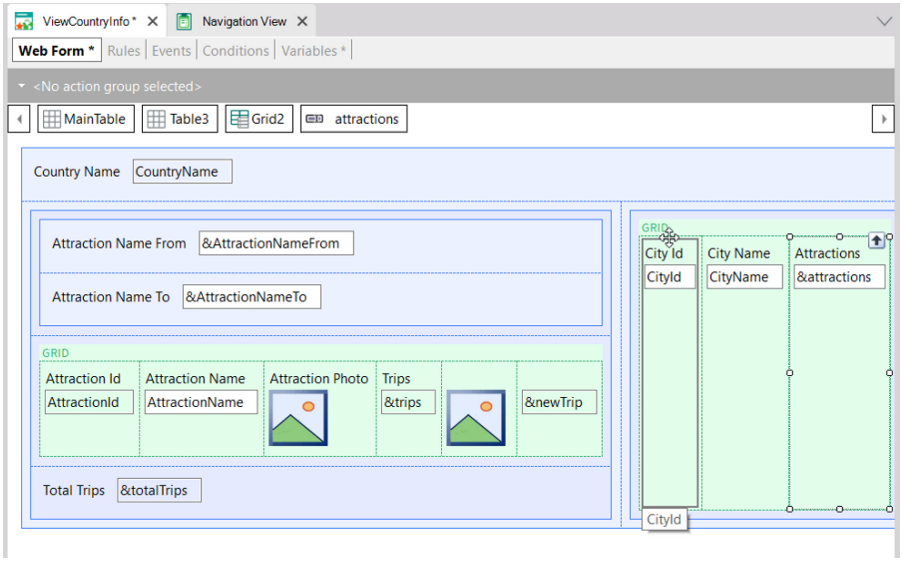
Novo grid



Executemos.

Trata-se de duas navegações independentes, mas como as duas têm relação com o país recebido por parâmetro, em ambas se está filtrando por país.

Novo grid



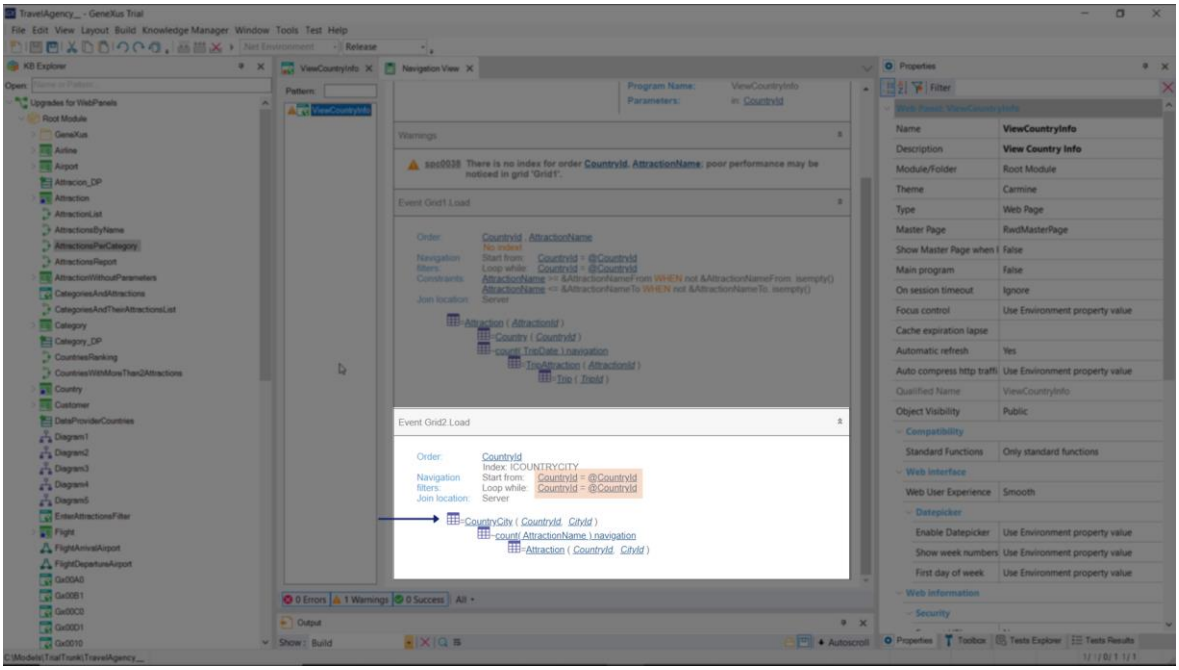

```
Event Grid1.Load
    &trips = Count(TripDate)
    &totalTrips = &totalTrips + &trips
Endevent

Event Grid2.Load
    &attractions = Count(AttractionName)
endevent


Event Refresh
    &totalTrips = 0
Endevent
```

Se assim como para as atrações calculamos a quantidade de trips, para as cidades quiséssemos calcular a quantidade de atrações que cada uma tem... então, adicionamos uma variável &attractions ao grid, e a calculamos cada vez que se vai carregar uma linha, isto é, no evento Load do grid de nome Grid2.

Por que não é necessário condicionar esta fórmula para que conte só as atrações do país e cidade?

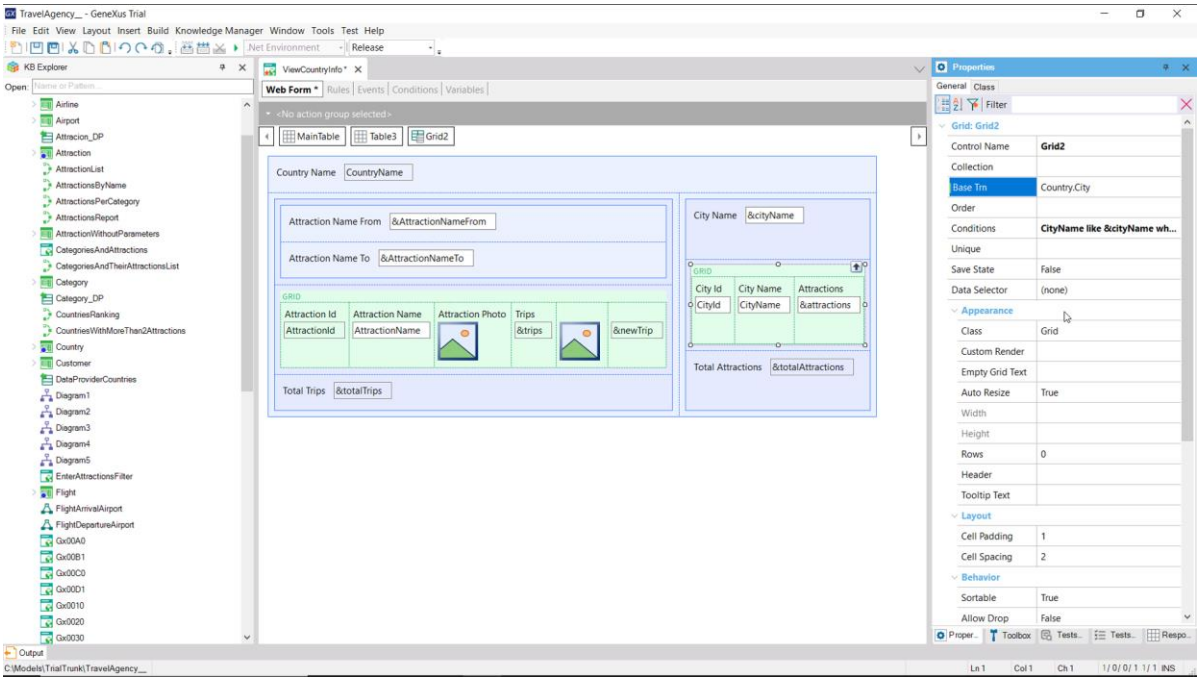
Executemos.

Novo grid



Enquanto gera, observemos a lista de navegação. Vemos que dentro do evento Load que será executado cada vez que se encontre um registro da tabela das cidades que corresponda ao país recebido por parâmetro, dispara-se o cálculo da fórmula count sobre Attraction, filtrando por país e cidade.

Novo grid



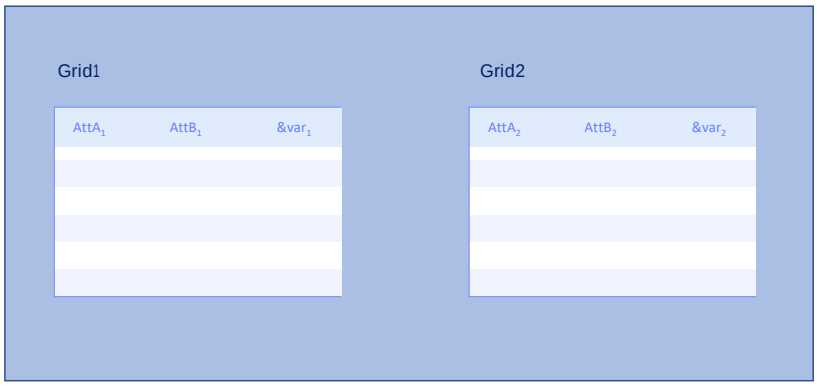
Para fazê-lo funcionalmente igual ao outro, podemos adicionar ao grid uma variável para filtrar as cidades mostradas e outra para mostrar o total de atrações de todas as cidades. Aqui já o fizemos.

Observemos que colocamos o filtro nas conditions do grid, utilizando o operador like. Não indicamos transação base e vimos que GeneXus a descobriu sozinho, mas nos convém fazê-lo.

Agora, tínhamos a inicialização da variável &totalTrips no evento Refresh genérico, e agora devemos inicializar também a variável &totalAttractions.

Mas temos, na verdade, três eventos Refresh: o genérico, que é o que temos por agora programado, e temos um Refresh de cada grid.

Ordem de execução dos eventos



Start

Refresh

Grid1.Refresh

Grid1.Load

Grid2.Refresh

Grid2.Load

A ordem de execução dos eventos ao executar o web panel pela primeira vez será:

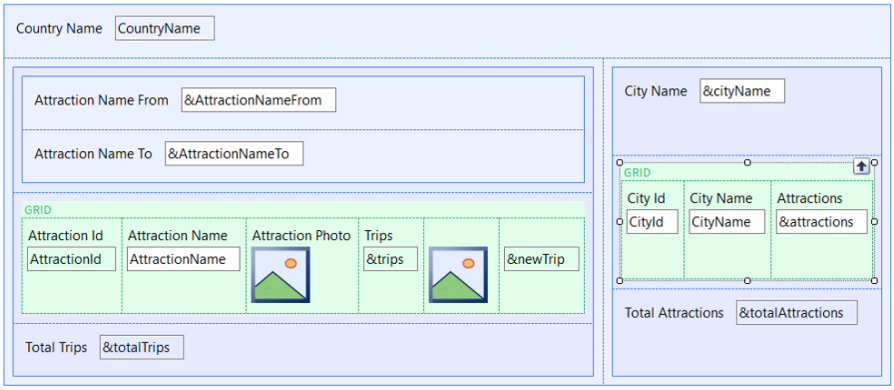
O evento Start

O evento Refresh genérico primeiro.

O Refresh do primeiro grid depois e, a seguir, se tiver tabela base, vai percorrer essa tabela filtrando os registros que correspondam, e executando o Load desse grid para cada um. Se não tem tabela base, então se executa o evento Load do grid apenas uma vez.

E depois, o mesmo com os eventos Refresh e Load do segundo grid.

Ordem de execução dos eventos




```
Event Grid1.Refresh
    &totalTrips = 0
Endevent

Event Grid2.Refresh
    &totalAttractions = 0
Endevent
```

No nosso exemplo, então, as variáveis &totalTrips e &totalAttractions deveriam inicializar-se no Refresh de cada grid, e não no genérico, porque depois, a ideia é que se mudarmos as variáveis de filtro de um grid, somente se atualize o que faz a esse grid, e não ao resto da tela.

Então, mudaríamos nossos eventos deste modo.

O que você deseja atualizar?

Start

Refresh

Grid1.Refresh
Grid1.Load

Grid2.Refresh
Grid2.Load

Event ´User-event´
…
Form.Refresh
endevent

Event ´User-event´
…
Refresh
endevent

Event ´User-event´
…
Grid1.Refresh()
endevent

Claro, o aparecimento de mais grids faz com que o comando Refresh que tínhamos visto em outra aula programado em um evento de usuário, possa especializar-se.

Por exemplo:

Temos o comando **Form.Refresh** que fará com que se atualize toda a página, executando-se Start, Refresh genérico, Refresh e Load de cada grid.

O comando **Refresh** genérico (o que tínhamos visto) faz com que se executem Refresh genérico, e Refresh e Load de cada grid (ou seja, tudo menos o Start).

E, agora, temos também o método Refresh de um grid, que fará com que se atualize só o grid, ou seja, que se executem o Refresh do grid e o Load do grid (uma vez ou n, dependendo de, se não tem ou tem tabela base).

Comando Load?

Start

Refresh

Grid1.Refresh
Grid1.Load

Grid2.Refresh
Grid2.Load

Event Grid1.Load
…
Load
endevent

Event 'User-event'
…
Grid1.Load()
endevent

Para o caso do comando Load a coisa muda um pouco.

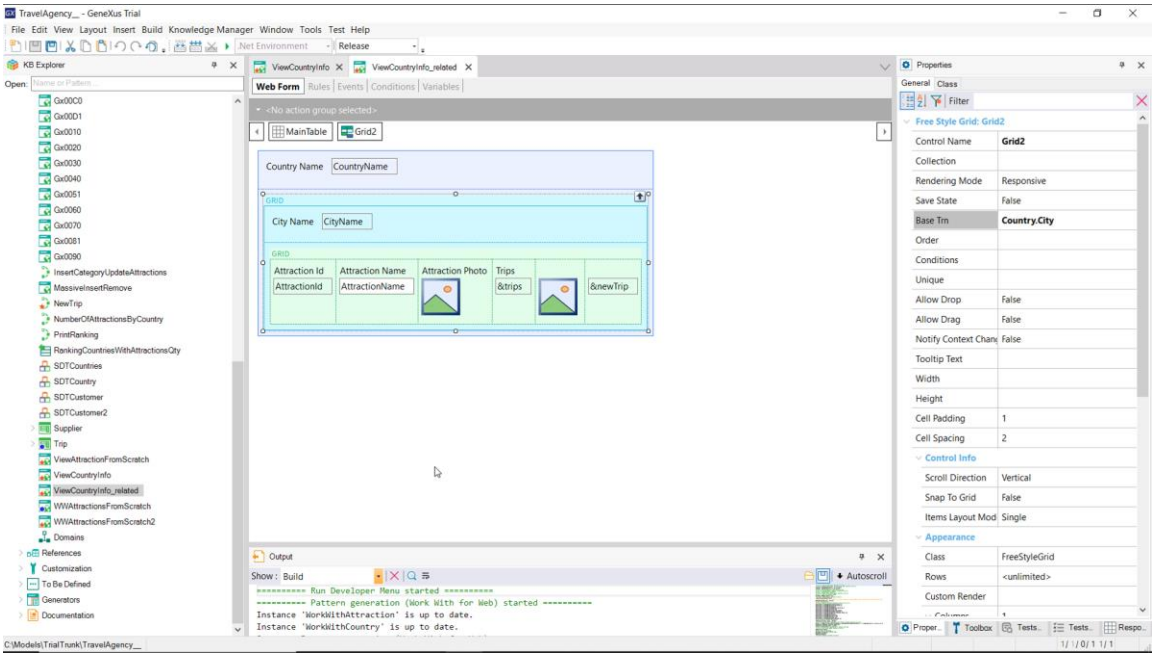
Quando houver mais de um grid, o **comando Load** só poderá ser escrito dentro do **evento Load** do grid de que se trata.

E se quiser carregar uma linha em um dos grids a partir de um evento de usuário, para isso terá que utilizar, necessariamente, o método Load do grid de que se trata.

Grids paralelos ou aninhados?

Aqui só vimos um exemplo de grids paralelos, mas os grids também podem aninhar-se, como os for eachs.

Grids aninhados: Free style grid

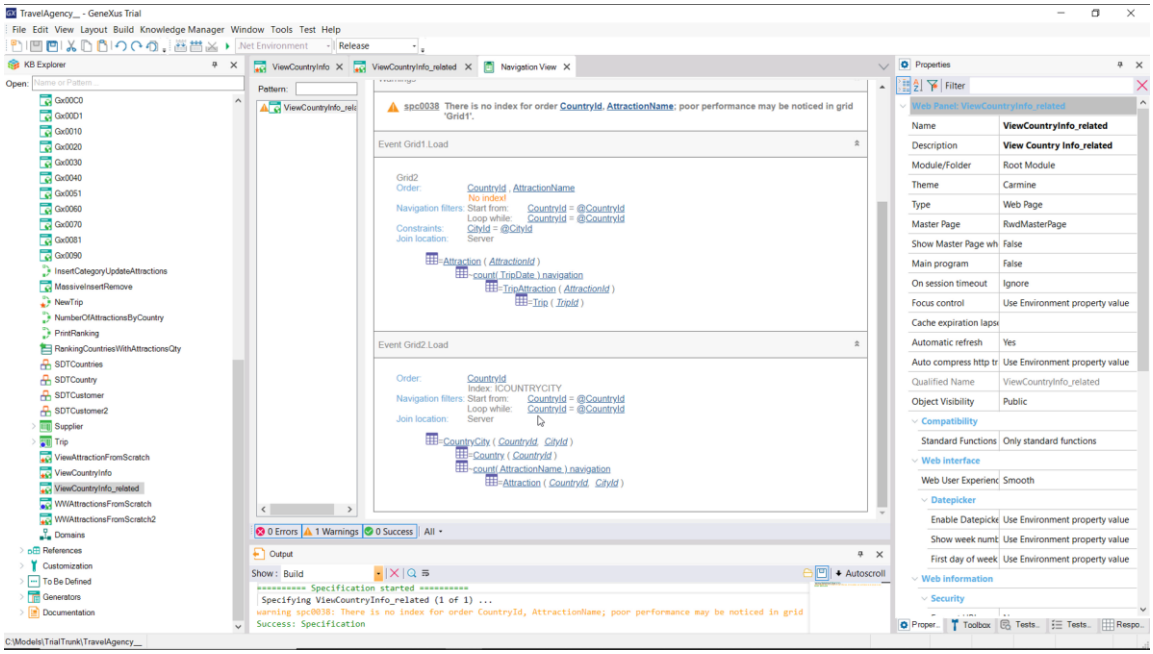


Por exemplo, se quiséssemos mostrar o país escolhido com a informação que vimos antes, mas de maneira relacionada.

Aqui o implementamos. Para que um grid possa conter outro, tem que ser um tipo especial de grid, de estilo livre e não tabular. Se chama Grid Freestyle.

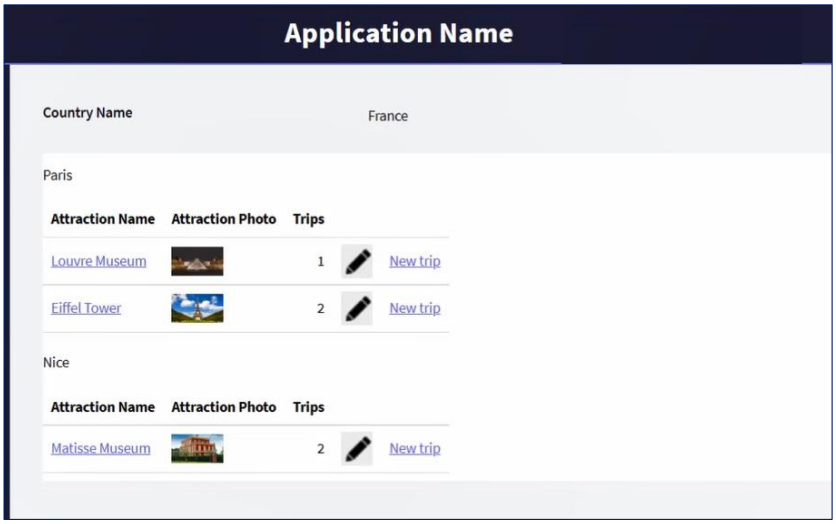
Este grid vai percorrer a tabela CountryCity e para cada cidade encontrada, vai executar o Refresh e Load do segundo grid –o grid aninhado- que vai buscar as atrações desse país-cidade. Aqui podemos ver como a informação está relacionada.

Grids aninhados: Free style grid



Se observamos a lista de navegação, vemos que o Grid2 que era o de cidades –está percorrendo essa tabela, CountryCity, filtrando por país recebido por parâmetro- e então, no Load do Grid1 que é o que corresponde às atrações, está sendo percorrida a tabela de atrações filtrando pelo país e a cidade na qual se está posicionado em cada registro de CountryCity e por isso está aparecendo este arroba daqui por CityId.

Grids aninhados: Free style grid



Se executamos. Aqui o vemos. Paris e suas duas atrações. Nice e sua atração.

Grids paralelos ou aninhados?

Há muito mais para pesquisar sobre este tema. Aqui, nos contentamos com esta introdução.

GeneXus™

training.genexus.com
wiki.genexus.com