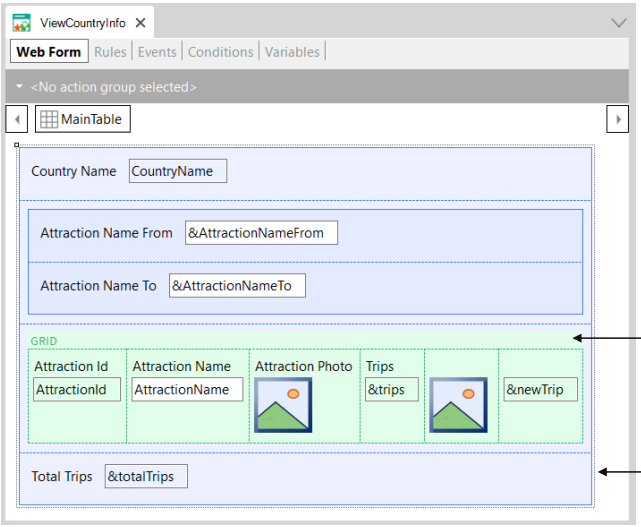
Web Screens with Back-office Focus

Web Panel Object. Multiple Grids.

GeneXus™

More than one grid



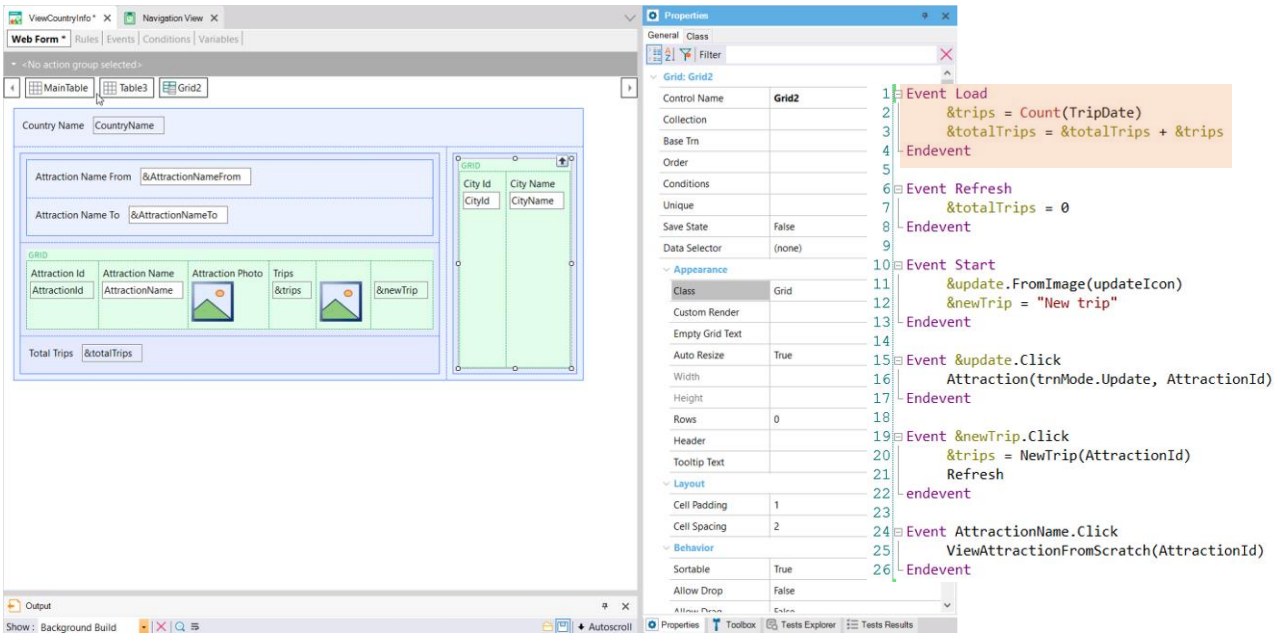
Grid1

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

+ Grid2

We've said a couple of times that maybe it would have been better to use the Load event of the grid and not the generic one, which only works in the case of a web panel without a grid or with a single grid. Using the Load event of the grid we anticipate a future need to enter another grid.

New grid



Let's suppose that in the web panel that shows the information of a country (its name and its tourist attractions), we want to also add a grid with its cities. Before doing so, note that its navigation list indicates the load of a single grid – for now.
Before adding the grid for the cities, let's insert in a table everything corresponding to the country's attractions, so that all that information is stored together.

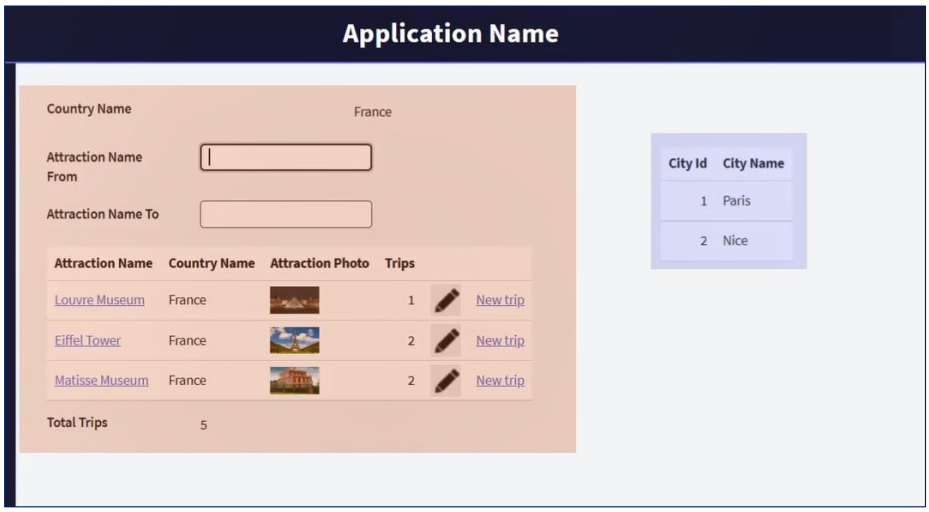
Next, let's insert another table for the city information. There we will insert the new grid, made up of the attributes CityId and CityName.
Looking at its properties, we see that it has been named Grid2 by default.

Each grid may or may not have a base table. In this case both grids have attributes, so both will have a base table. How do you know which one the generic Load event code applies to? In fact, if we save, we see that the navigation list shows a warning error.

It is showing the navigations that will have to be made to load each grid, and it even understood that the formula to calculate the trips must belong to the Grid1 loading, but we are asked to specify this. And we will do so.

Now, when saving the object, the navigation list no longer shows the error.

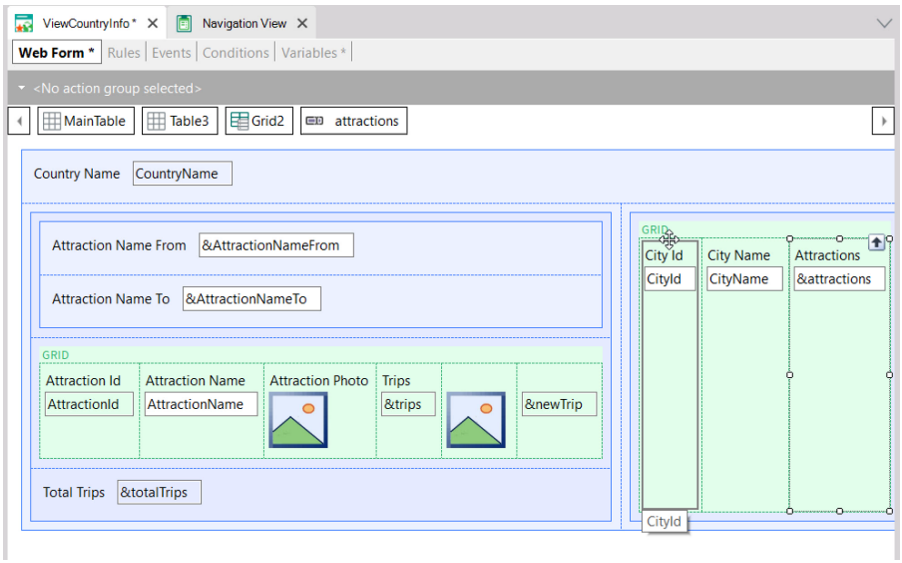
New grid



We run it...

These are two independent navigations, but since both are related to the country received by parameter, in both of them it is filtering by country.

New grid



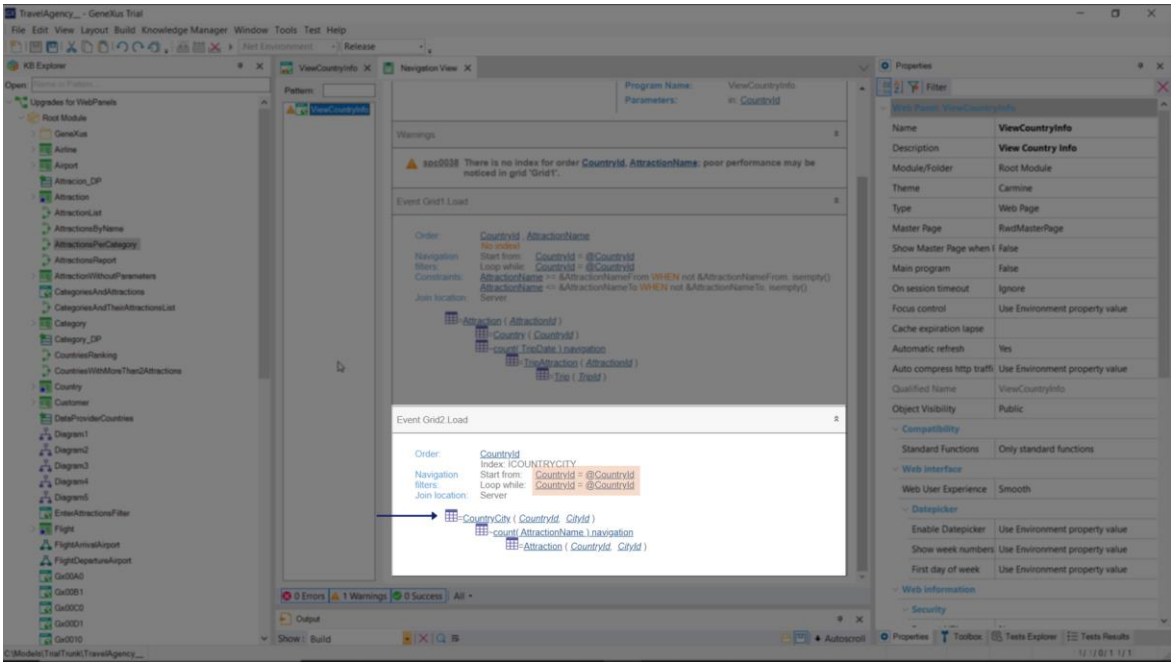

```
Event Grid1.Load
    &trips = Count(TripDate)
    &totalTrips = &totalTrips + &trips
Endevent

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


Event Refresh
    &totalTrips = 0
Endevent
```

If, like for the attractions we calculate the number of trips, for the cities we would like to calculate the number of attractions that each one has... then we add a variable &attractions to the grid, and calculate it every time a line is going to be loaded; that is, in the Load event of the grid named Grid2.

Why is it not necessary to condition this formula to count only the attractions of the country and city?

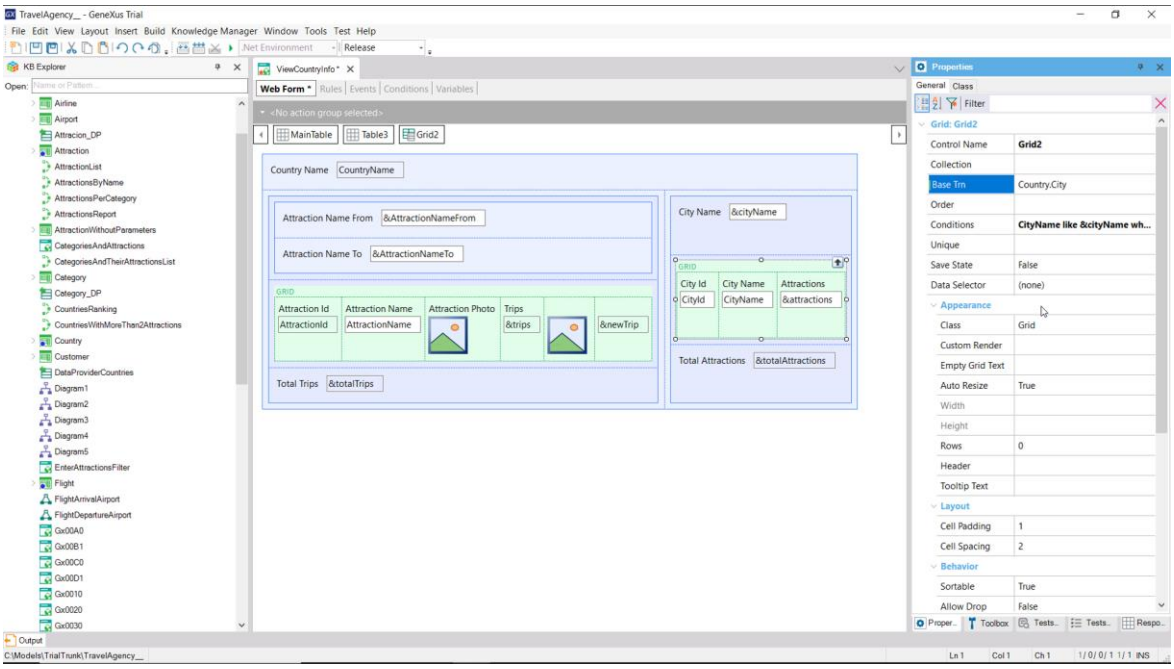
We run it...

New grid



While generating, let's look at the navigation list. Within the Load event that will be executed every time a record is found in the table of cities corresponding to the country received by parameter, the calculation of the Count formula is triggered on Attraction, filtering by country and city.

New grid



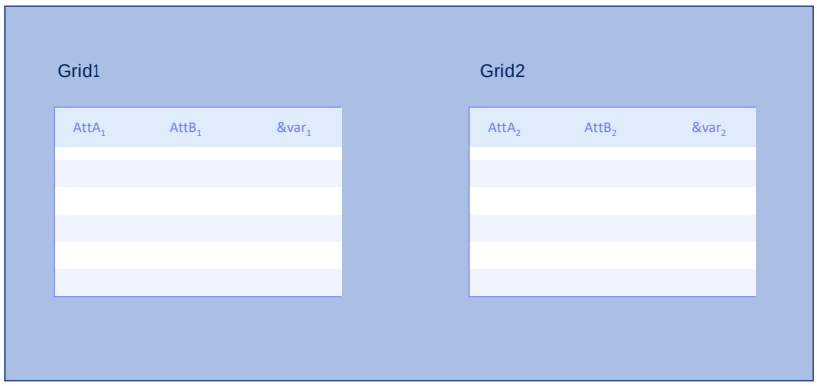
To make it functionally the same as the other, we can add a variable to the grid to filter the cities shown and another to show the total of attractions of all the cities. We have done it here.

Note that we place the filter in the grid conditions, using the like operator. We did not indicate a base transaction and saw that GeneXus discovered it on its own, but it is in our interest to do so.

We had initialized the &totalTrips variable in the generic Refresh event, and now we must also initialize the &totalAttractions variable.

But we have, in fact, three Refresh events: the generic one, which is the one we have programmed for now, and we have a Refresh of each grid.

Event execution order



Start

Refresh

Grid1.Refresh

Grid1.Load

Grid2.Refresh

Grid2.Load

The order of execution of the events when the web panel is executed for the first time will be:

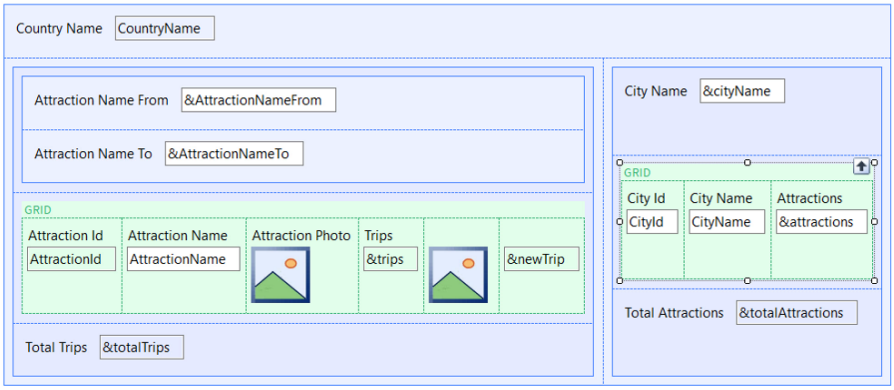
Start event

Generic Refresh event first.

Next, the Refresh of the first grid and then, if it has a base table, that table will be run through by filtering the corresponding records, and executing the Load event of that grid for each one. If it doesn't have a base table, then the Load event of the grid is executed only once.

And then the same with the Refresh and Load events of the second grid.

Event execution order




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

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

In this example, the variables &totalTrips and &totalAttractions should be initialized in the Refresh event of each grid, and not in the generic one. The reason is that if later on we need to change the filter variables of a grid, we may refresh only that grid and not the rest of the screen.

Then we would change our events in this way.

What do you want to refresh?

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

Of course, the addition of more grids means that the Refresh command that we had seen in another class programmed in a user event can be specialized.

For example:

We have the command **Form.Refresh** that will cause the entire page to be refreshed, executing Start, generic Refresh, Refresh and Load for each grid.

The generic **Refresh** command (that we had seen) causes the generic Refresh, and Refresh and Load of each grid (i.e. everything but Start) to be executed.

And now we also have the Refresh method of a grid, which will refresh only the grid; that is to say, run the grid Refresh and Load (once or n times, depending on whether it has a base table or not).

Load command?

Start

Refresh

Grid1.Refresh

Grid1.Load

Grid2.Refresh

Grid2.Load

Event Grid1.Load

…

Load

endevent

Event 'User-event'

…

Grid1.Load()

endevent

As for the Load command, things are a little different.

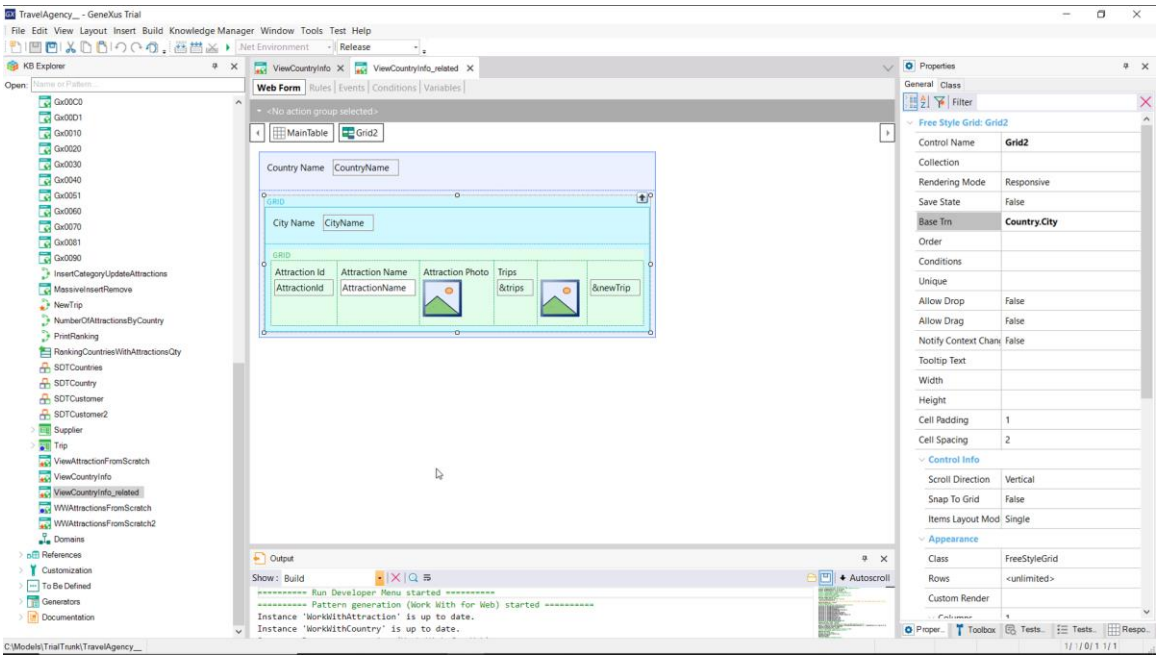
When there is more than one grid, the **Load command** alone can only be written within the **Load event** of the grid in question.

And to load a line in one of the grids from a user event, you will have to use, necessarily, the Load method of the grid in question.

Parallel or Nested Grids?

Here we've only seen one example of parallel grids, but grids can also be nested, like For Each commands.

Nested grids: Free style grid

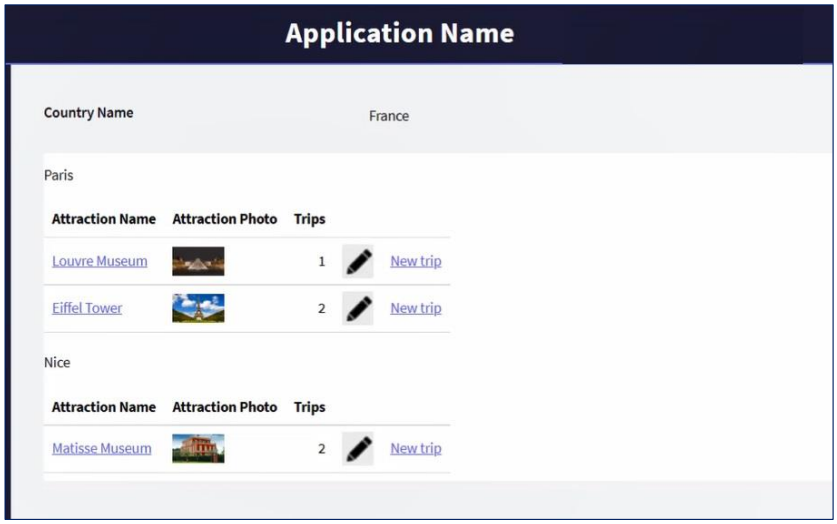


For example, if we wanted to show the selected country with the information we saw before, but in a related way.

We implement it here. For one grid to contain another, it has to be a special type of grid, freestyle and not tabular. It is called Freestyle Grid.

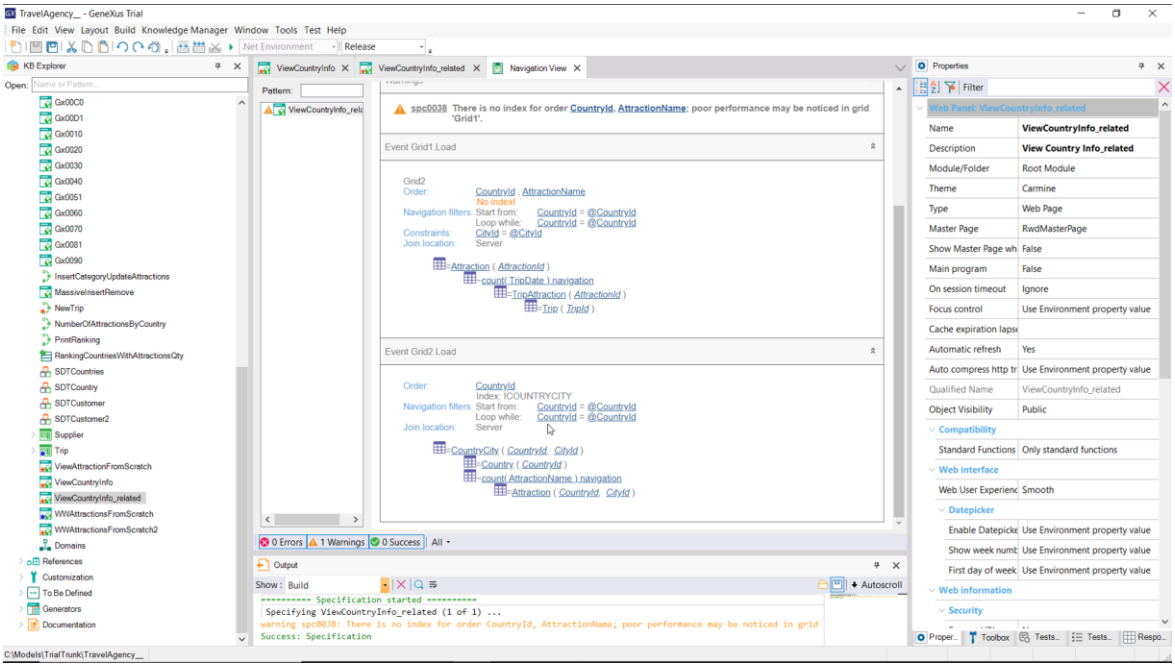
This grid will run through the CountryCity table and for each city found, it will run the Refresh and Load of the second grid –the nested grid– which will search for the attractions of that country-city. There we can notice how the information is related.

Nested grids: Free style grid



If we run it, we can see it here. Paris and its two attractions. Nice and its attraction.

Nested grids: Free style grid



In the navigation list, we see that Grid2 –the cities one– is running through that table, CountryCity, filtering by the country received by parameter. Then, the Load of Grid1, which corresponds to the attractions, is running through the attractions table filtering by the country and the city in which it is positioned in each record of CountryCity, and that's why this at sign here is displayed for CityId.

Parallel or Nested Grids?

There is much more to explore about this topic. Here we will only see this introduction.

GeneXus™

training.genexus.com
wiki.genexus.com