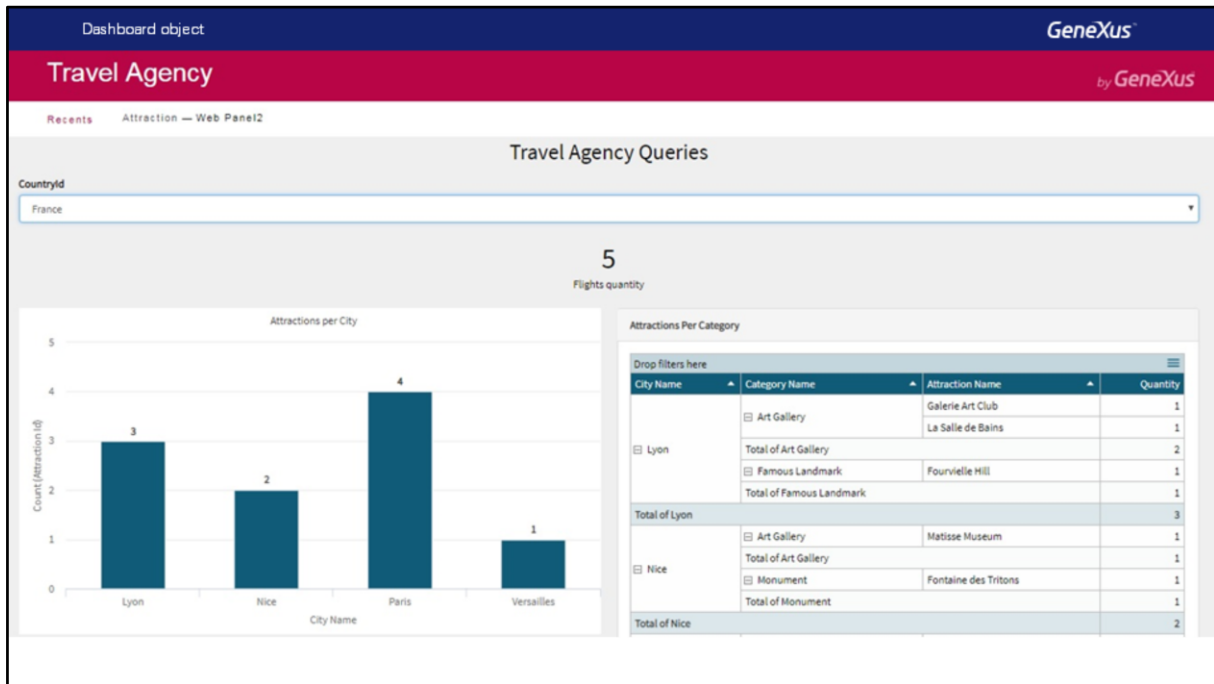


Designing dynamic queries

Dashboard object

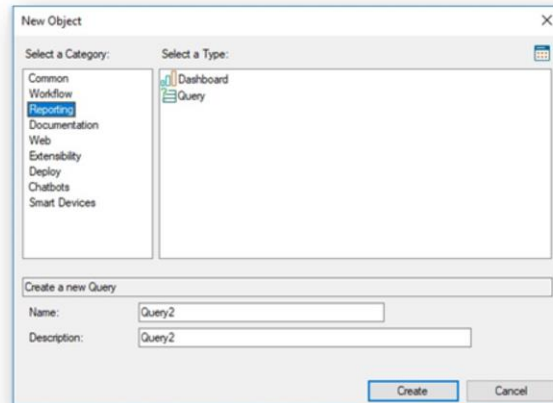
GeneXus 16



We have previously seen how important and necessary it is to be able to query the database to analyze the information and make decisions.

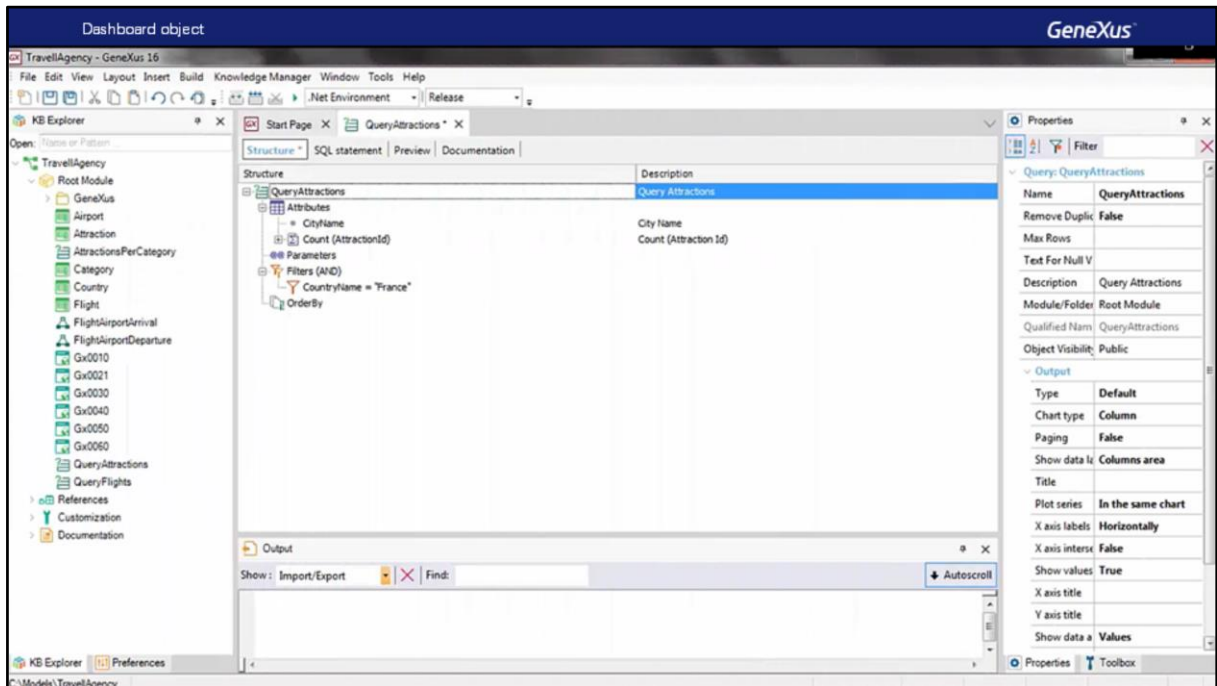
In this context, we may need to view several indicators together. For example, select a country and see on the same screen the number of flights registered to that country, a chart with the number of tourist attractions offered by each city in that selected country, and a dynamic table with the attractions grouped by category, such as museums, monuments, art galleries, and so on.

Dashboardobject



The Dashboard object will allow us to meet this requirement, and show all the indicators on the same screen.

Thus, this object provides views of the so-called “Key performance indicators,” which are so important for a business process.



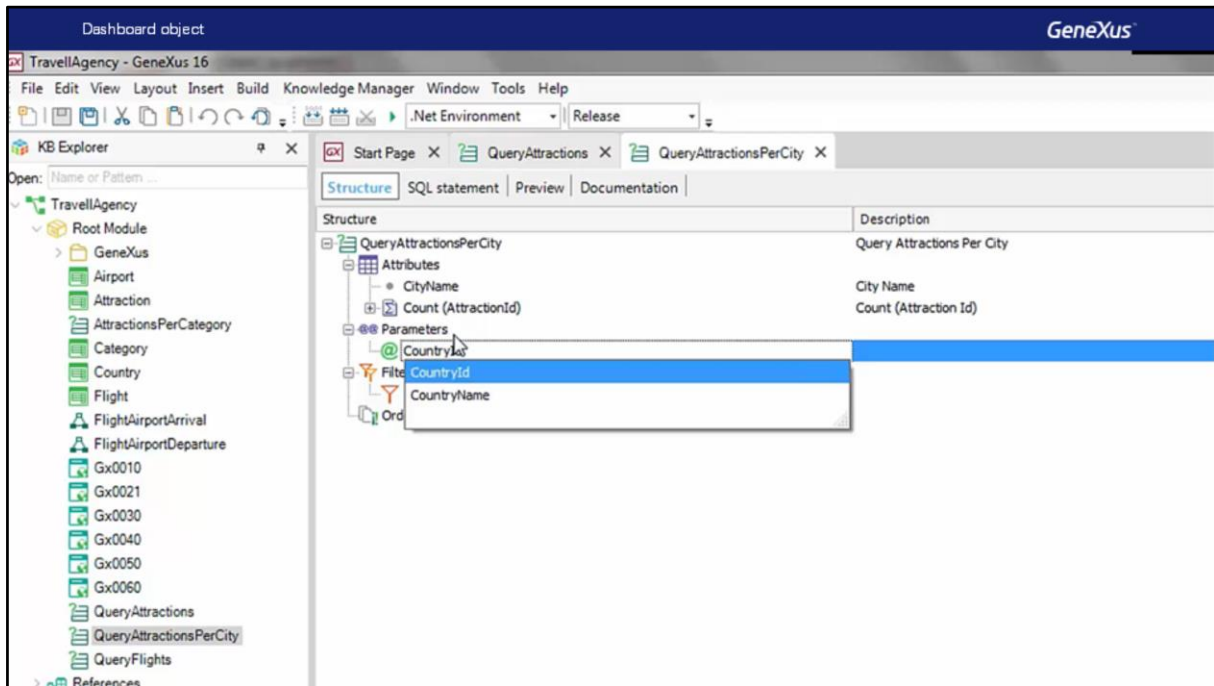
Let's see it in GeneXus.

In the previous video, we created a Query object to view a chart with the number of tourist attractions in France.

We will make this object "general" in order to receive a certain country in a parameter and have the chart show the number of tourist attractions in each city of that country.

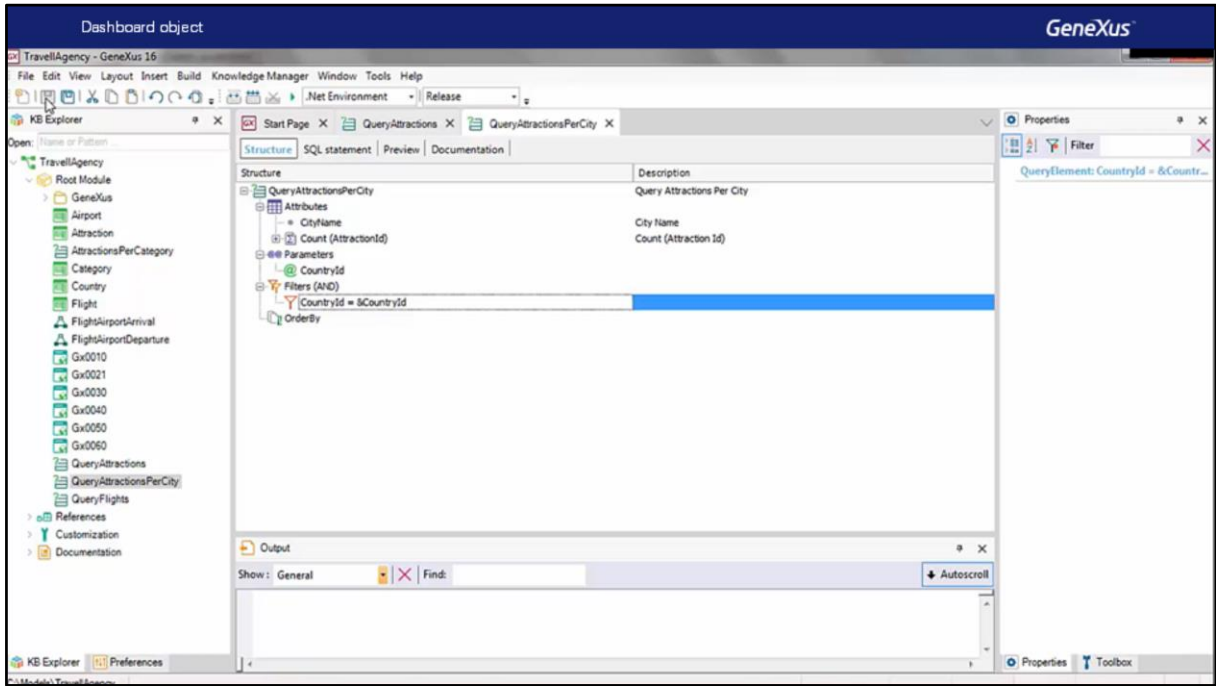
In order not to alter the requirements already defined, we will select "Save as" for this object... and call it QueryAttractionsPerCity.

Good. Now we will indicate that this query will receive a country identifier as a parameter.

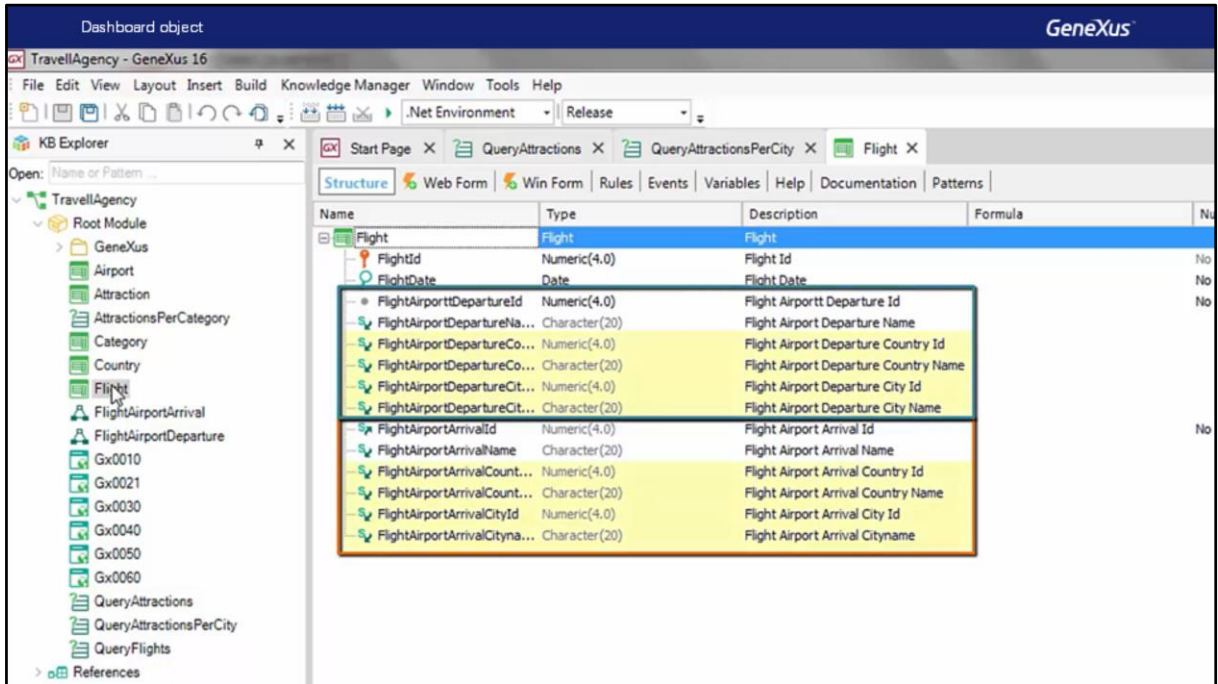


On the Parameters node, we press Enter, and declare the parameter is CountryId... We also change the definition:

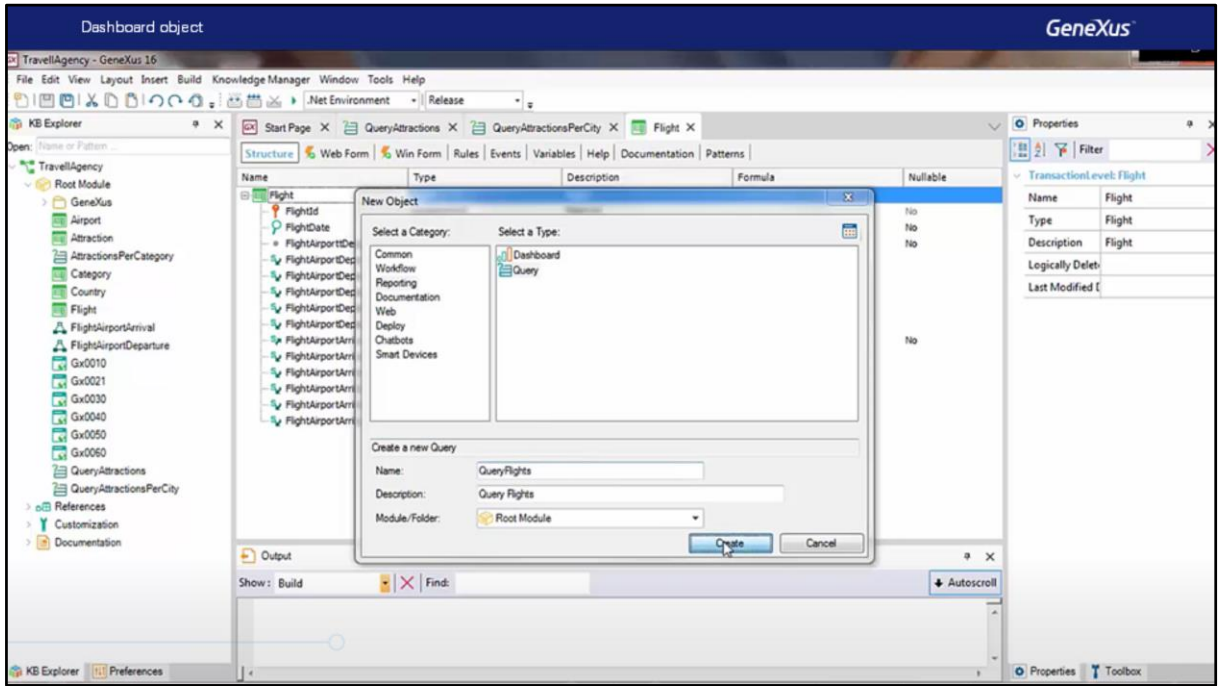
We already have a query that returns the number of tourist attractions for each city of a certain country received in a parameter.



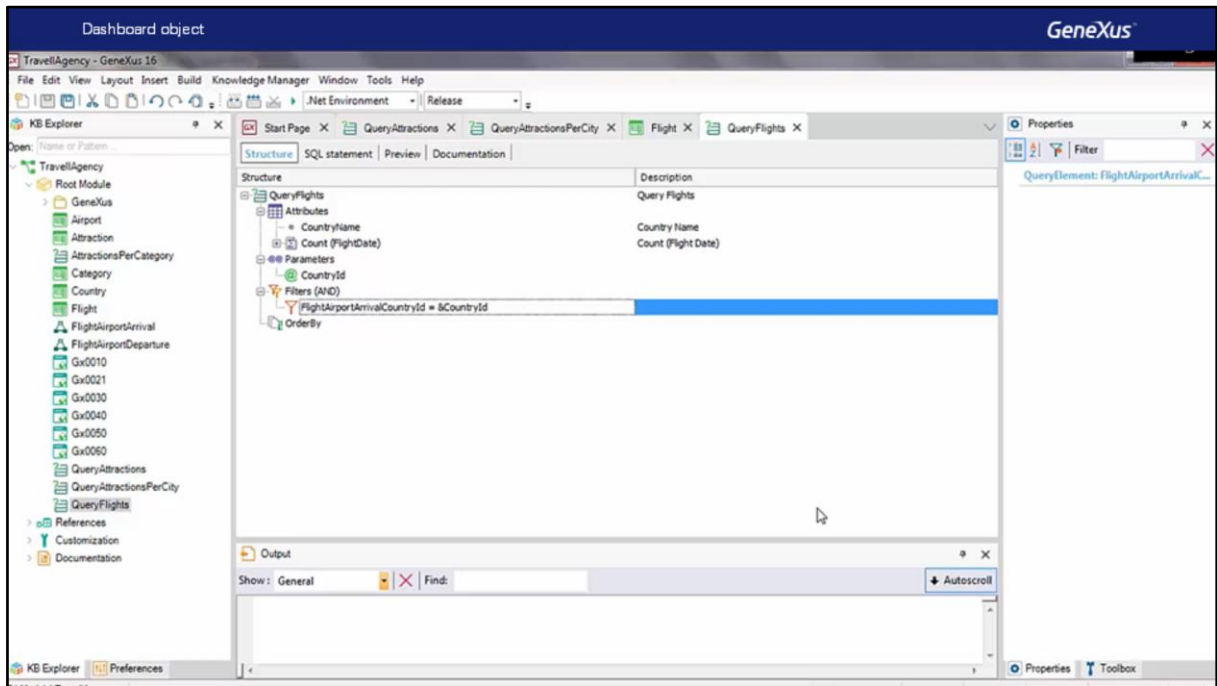
Now we will create another Query object so that it returns the total of flights registered for a certain country received in a parameter.



Let's first review the structure of the Flight transaction. We had indicated that a flight has one airport of arrival and one airport of departure. Note that to the corresponding groups of subtypes we have added the country and the city where each airport is located.



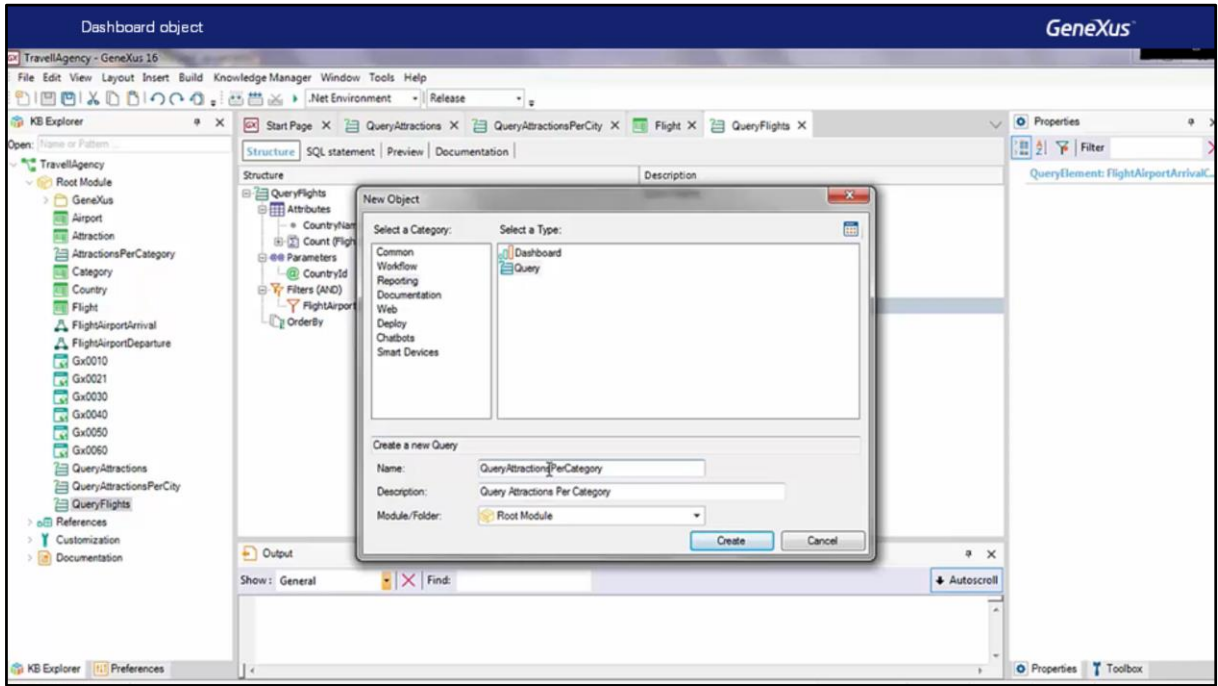
So, we create a new Query object and call it QueryFlights.



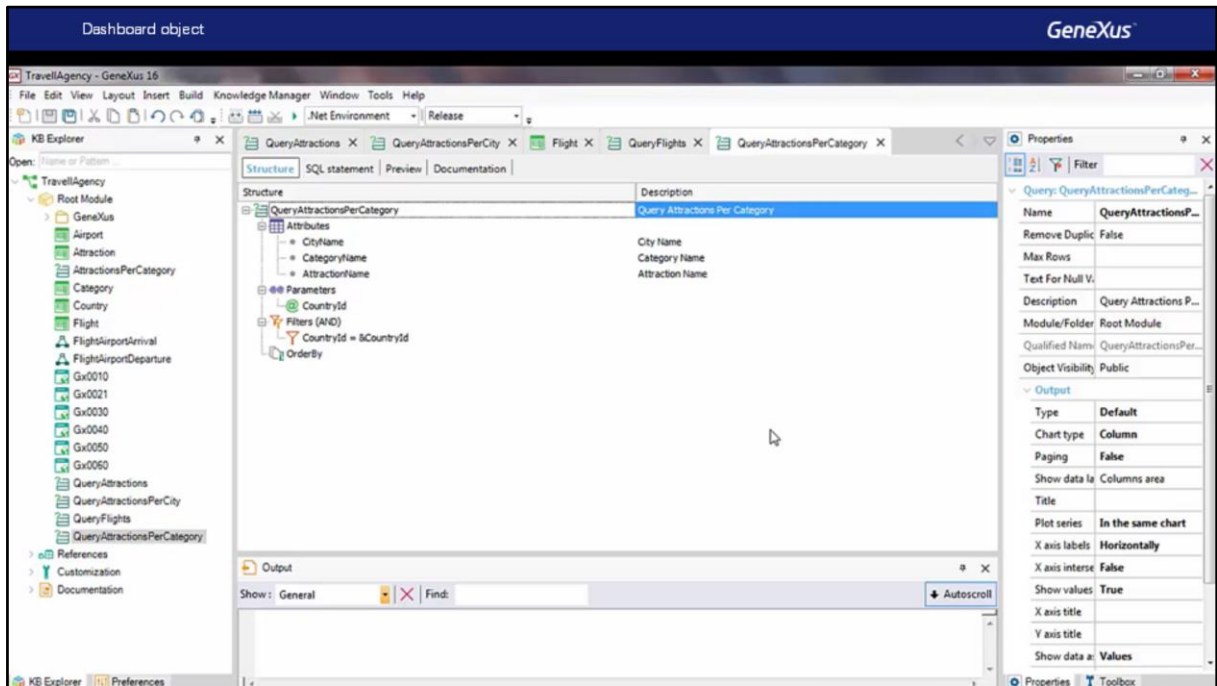
In the attributes node we declare CountryName and the number of flights to this country.

We state that the CountryId value is received as a parameter and also declare the corresponding filter condition.

Now, we will create the last Query object to obtain the list of tourist attractions grouped by category (such as museums, monuments, art galleries, etc.), for all the cities of a certain country also received in a parameter.



So, we create this last Query object...

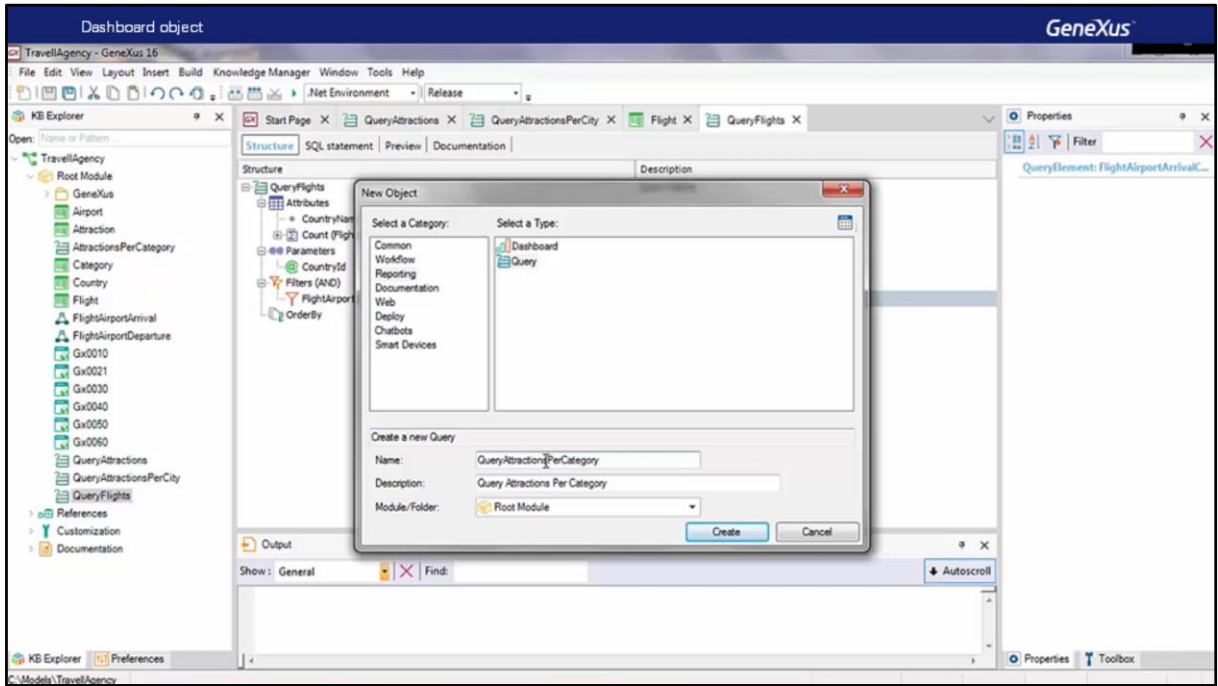


And in the Attributes node we declare CityName, CategoryName, and AttractionName. We use this order because it is the grouping order we want to see: First the cities, then the category names, and lastly the name of the tourist attractions.

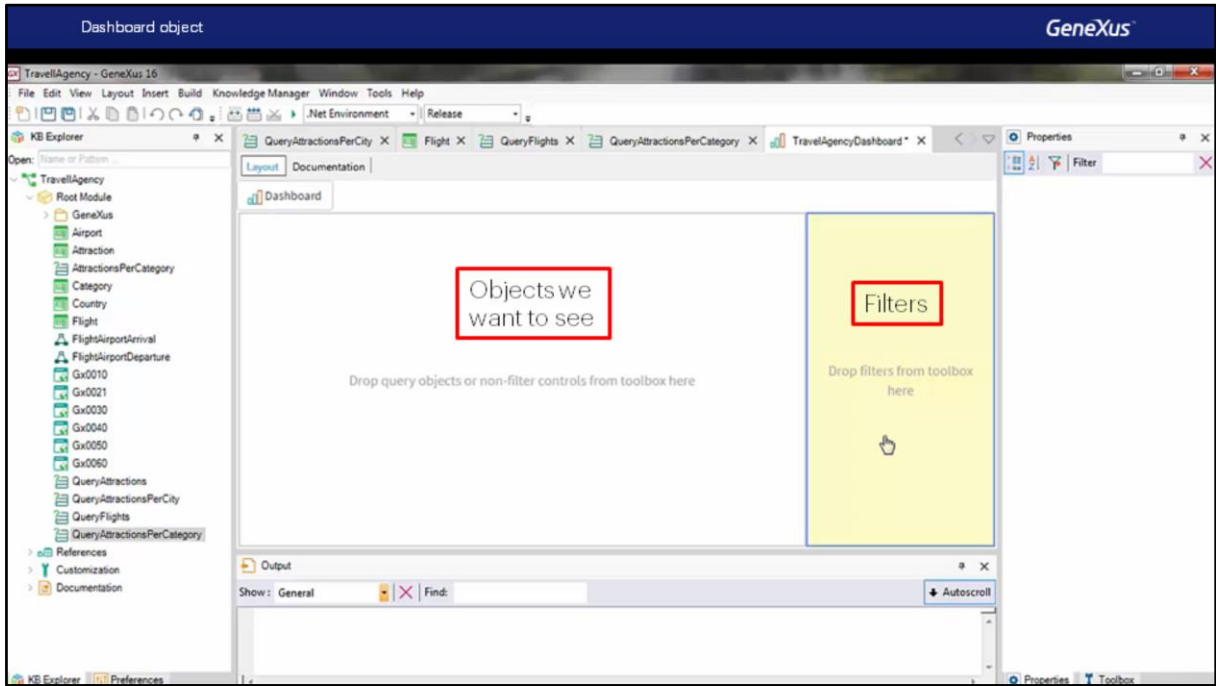
Finally, just like in previous queries, we declare the CountryId parameter and the corresponding filter condition.

At this point, we have three Query objects that solve different queries, and that also have something in common: they all receive a country identifier as a parameter.

Now we want to be able to query these three indicators at the same time in the same screen, so we will create an object of Dashboard type.



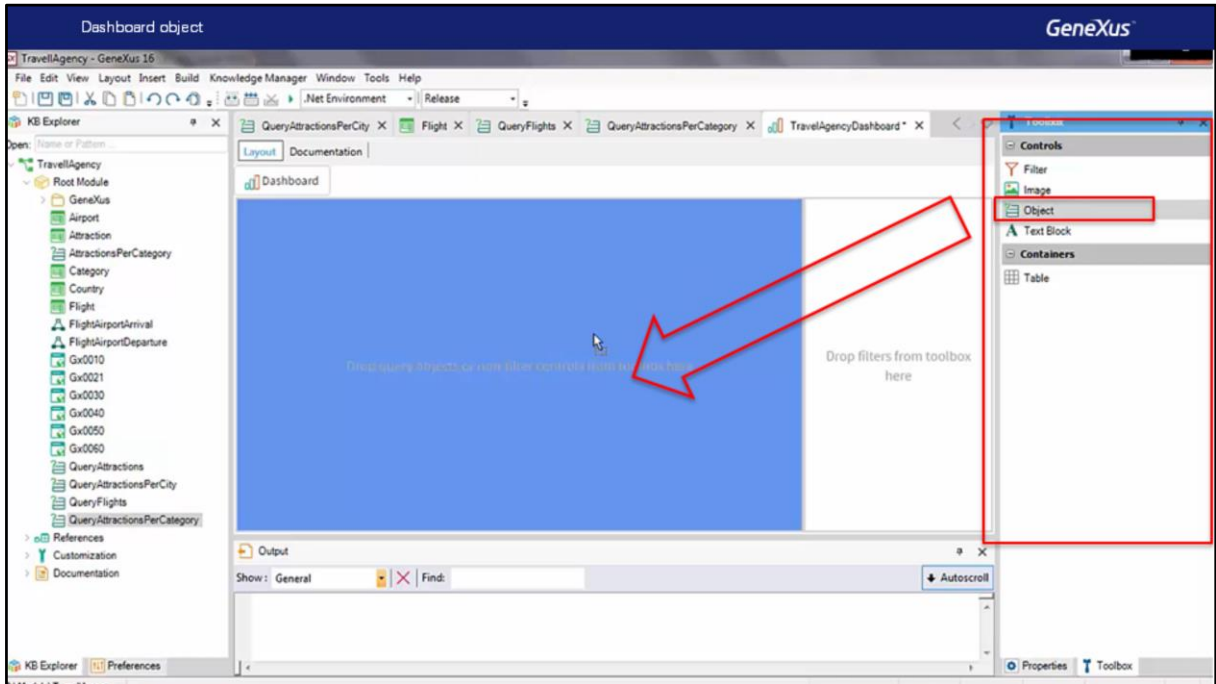
And we will call it TravelAgencyDashboard.



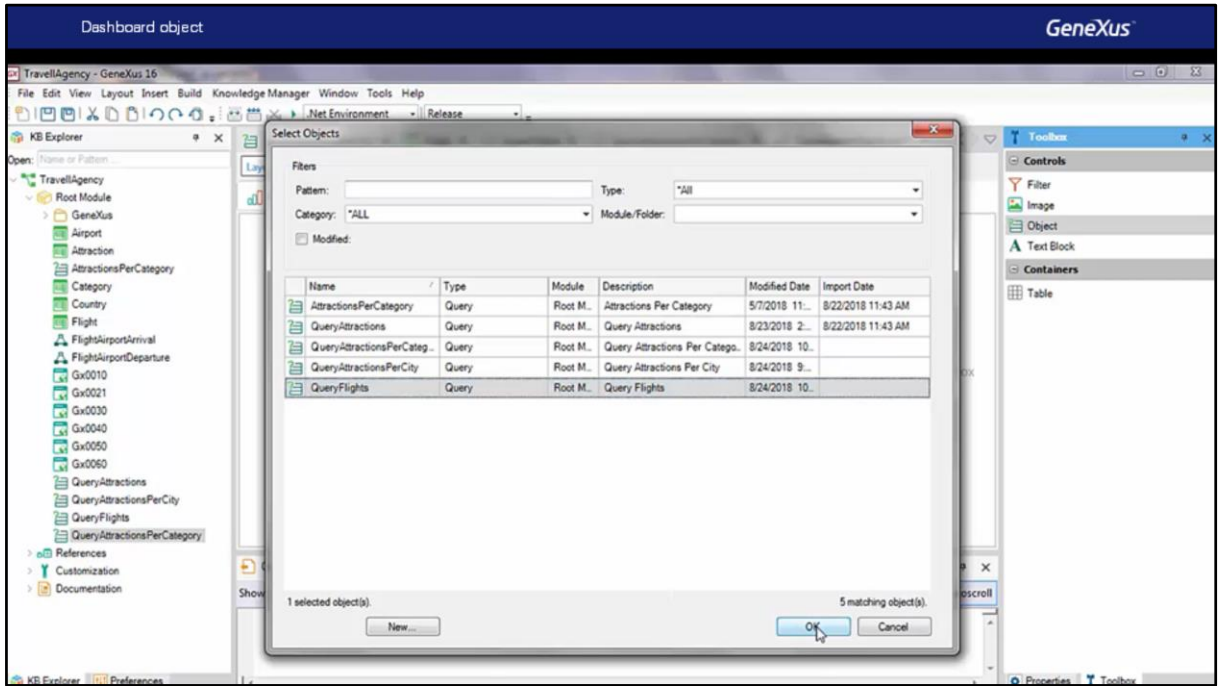
We can see two large areas:

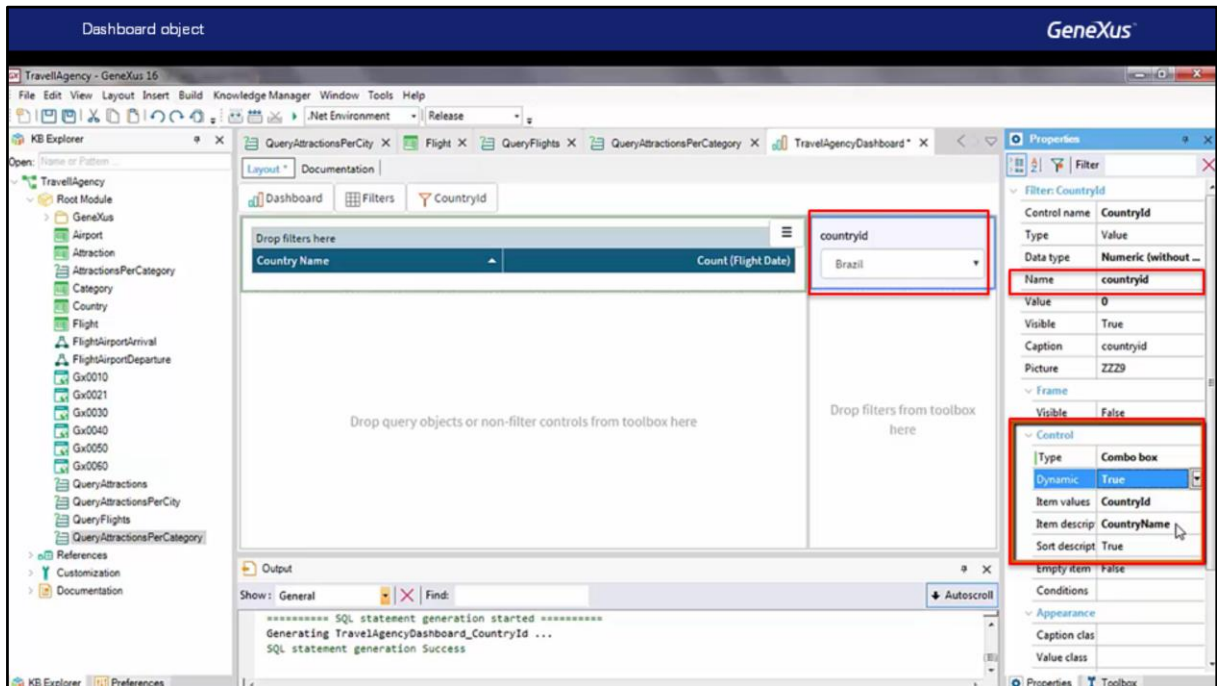
Here we drag the objects we want to see,

and here we define the corresponding filters:



We open the Toolbox, drag an Object control over the corresponding area, and select QueryFlights.

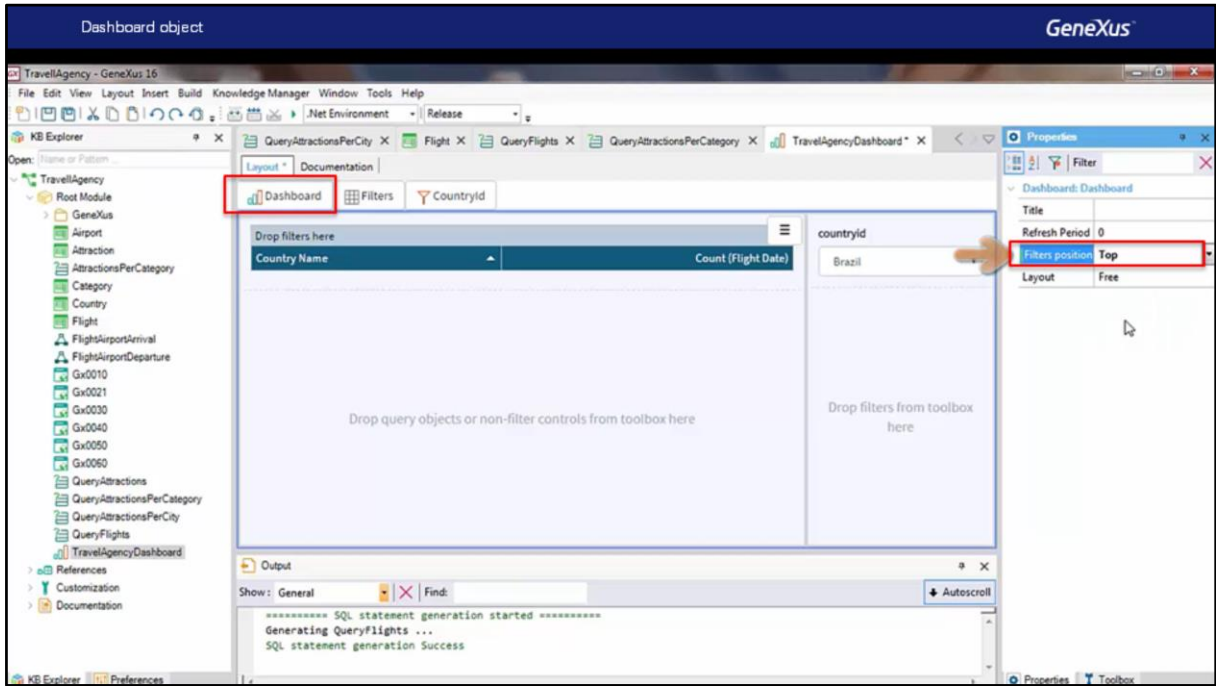




Note that since it is a Query object that has parameters and filters defined, this filter to define in the corresponding area of the Dashboard is automatically displayed.

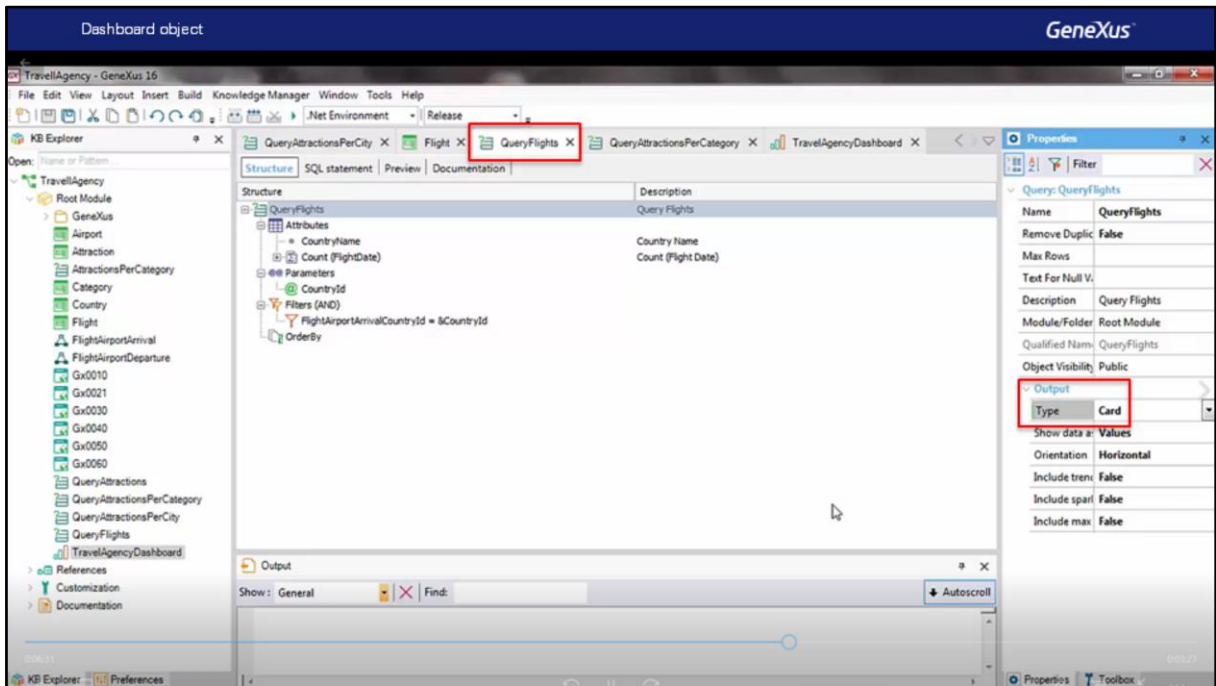
In its properties, we indicate the filter name CountryId, and also indicate that it is a dynamic combo.

From the Type property we select Combo Box, and set to True the Dynamic property. We also indicate that the value will be CountryId and the description will be CountryName.



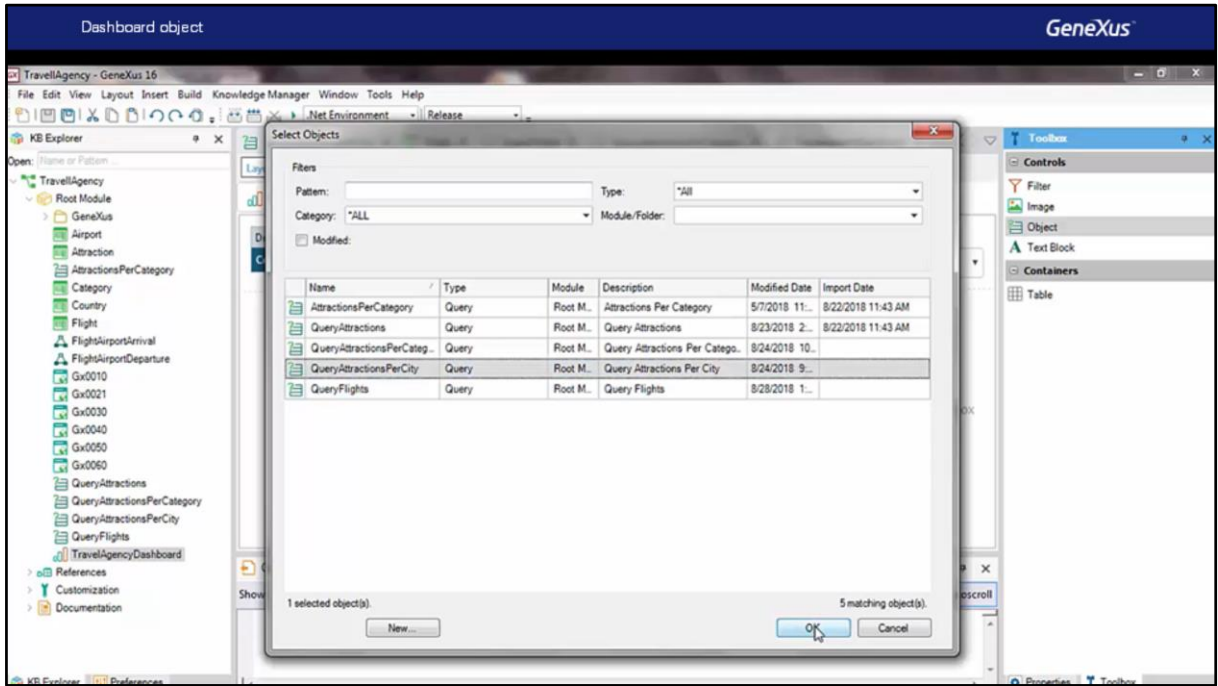
By clicking on this Dashboard tab we can see that its properties allow changing the position of the filters.

We choose Top:



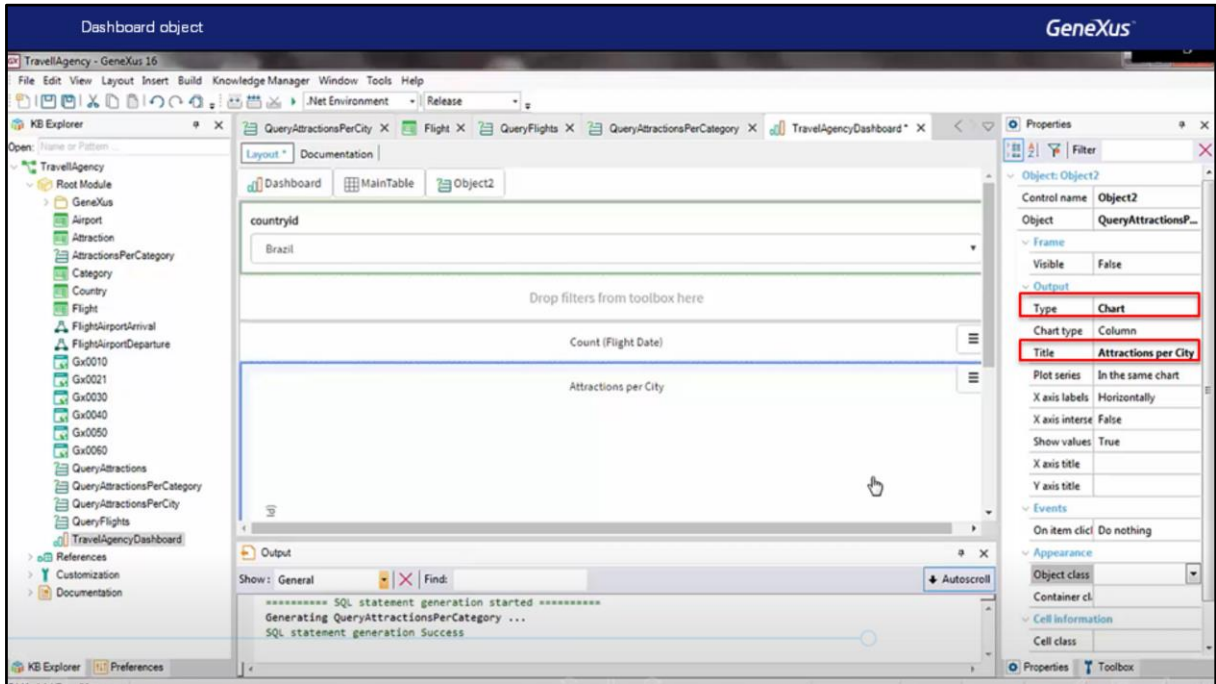
Good, we have defined the filter. Now we return to the QueryFlights object, and in the Type property, within the Output group, we set its output as Card.

This Card output format will allow us to see the number of flights registered for the country selected in the filter.



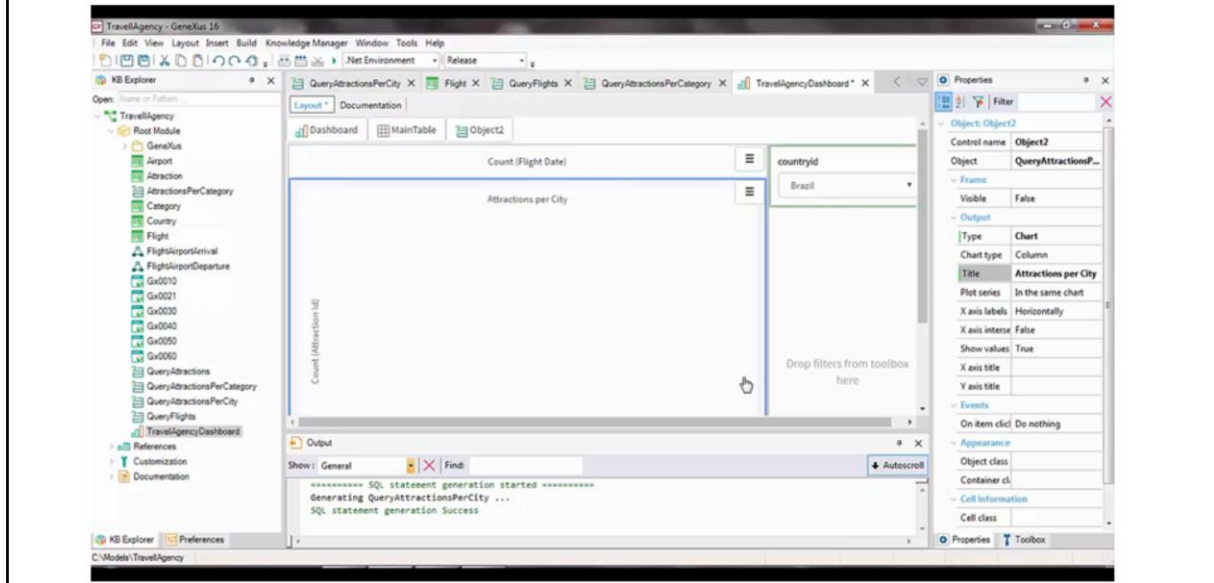
Let's return to the Dashboard object, and drag another control of Object type.

This time we select QueryAttractionsPerCity.



From its properties, we set the output as Chart and call it "Attractions per City."

DEMO: Continue Dashboard definition and execute ...



[DEMO: <https://youtu.be/oWJ0v4kDEyE>]

We repeat the process to integrate the last Query object:
From the Toolbox we drag the Object control again, and this time we choose QueryAttractionsPerCategory.

Within the Frame group we will also set this Visible property to True, and in this way we can enter a name for this Pivot Table. We will call it Attractions per Category.
Good, we will also give a title to the Query with Card output, so we select it and set to True its Visible property within the Frame group, and enter Query Flights as its title.

Now we will edit the filter properties and as a Caption we just enter "Country."

Let's consider the following:

By clicking on any of the Query objects we have in this Dashboard, the related filter is automatically highlighted. In our example, the three Query objects share the same filter, and that's why it is always highlighted.

Likewise, if we click on the filter, we can see that all the Query objects to which it applies are automatically highlighted.

At this point, to view this Dashboard at runtime we need to define a web panel and also use the Dashboard Viewer control.

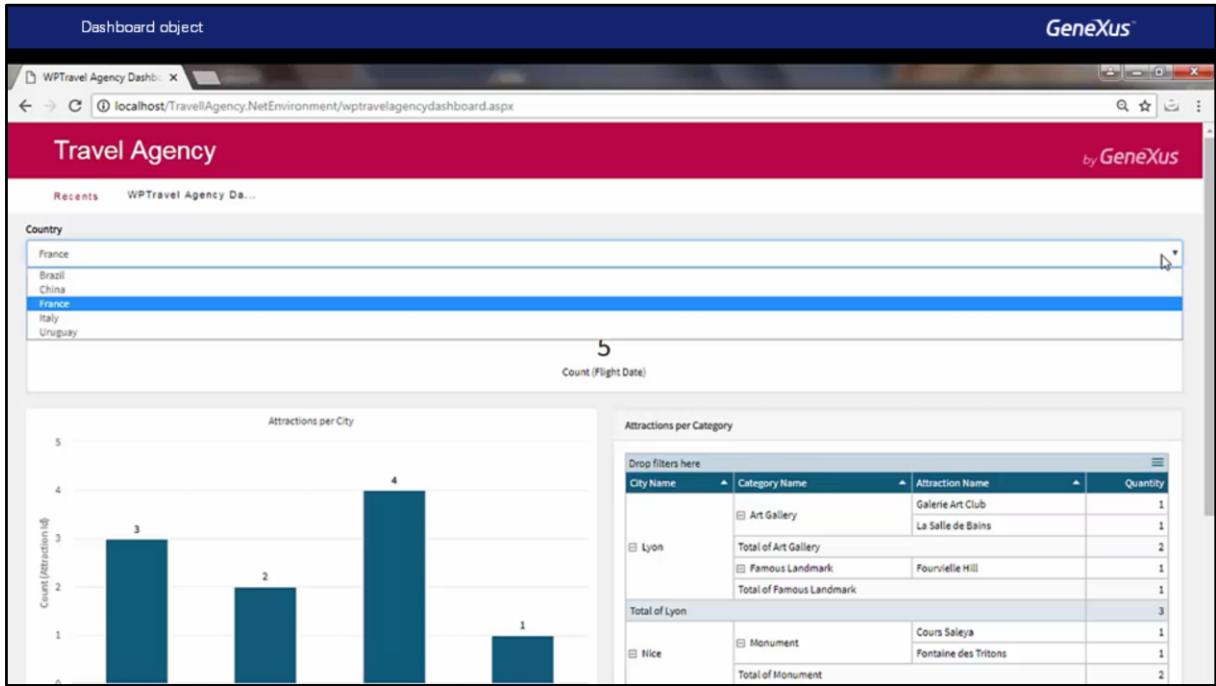
So, we create a web panel and call it WPTravelAgencyDashboard. From the Toolbox we drag the Dashboard Viewer control.

We edit its properties, and in the Object property we select TravelAgencyDashboard.

All we have to do now is press F5 to view our Dashboard running.

We select the web panel, and Brazil is displayed as the country in the filter; for example, we select France,

We change it to Italy... and then change it to China.



In this way, we have seen a simple example of the use of the Dashboard object to view several dynamic queries on the same screen receiving data from the same filter.

GeneXus™

The power of doing.

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