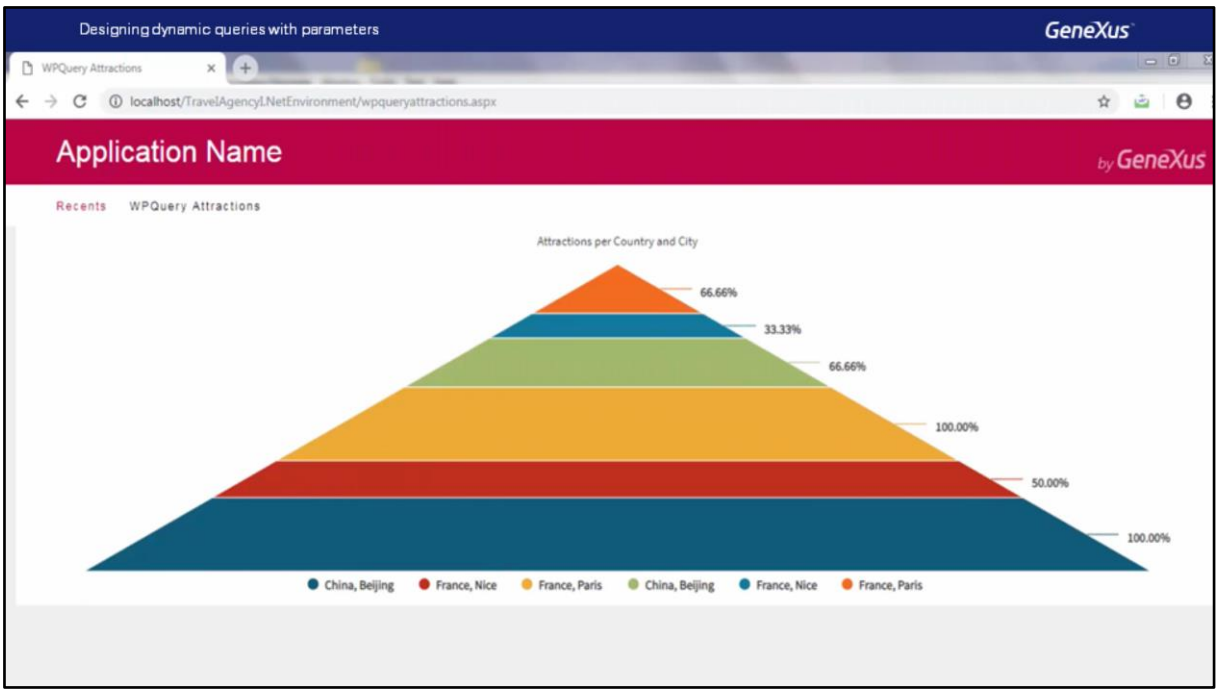
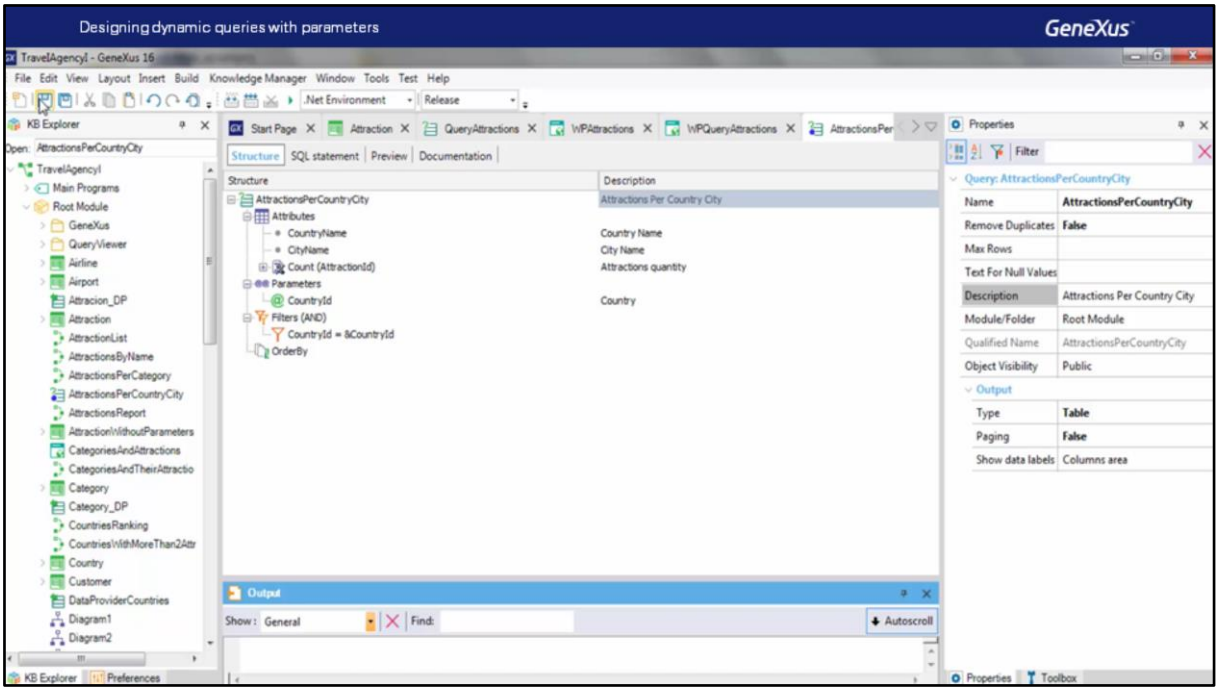
Designing dynamic queries with parameters

GeneXus 16



Previously, we had already defined a query object in our KB to view the number of tourist attractions in France and China.

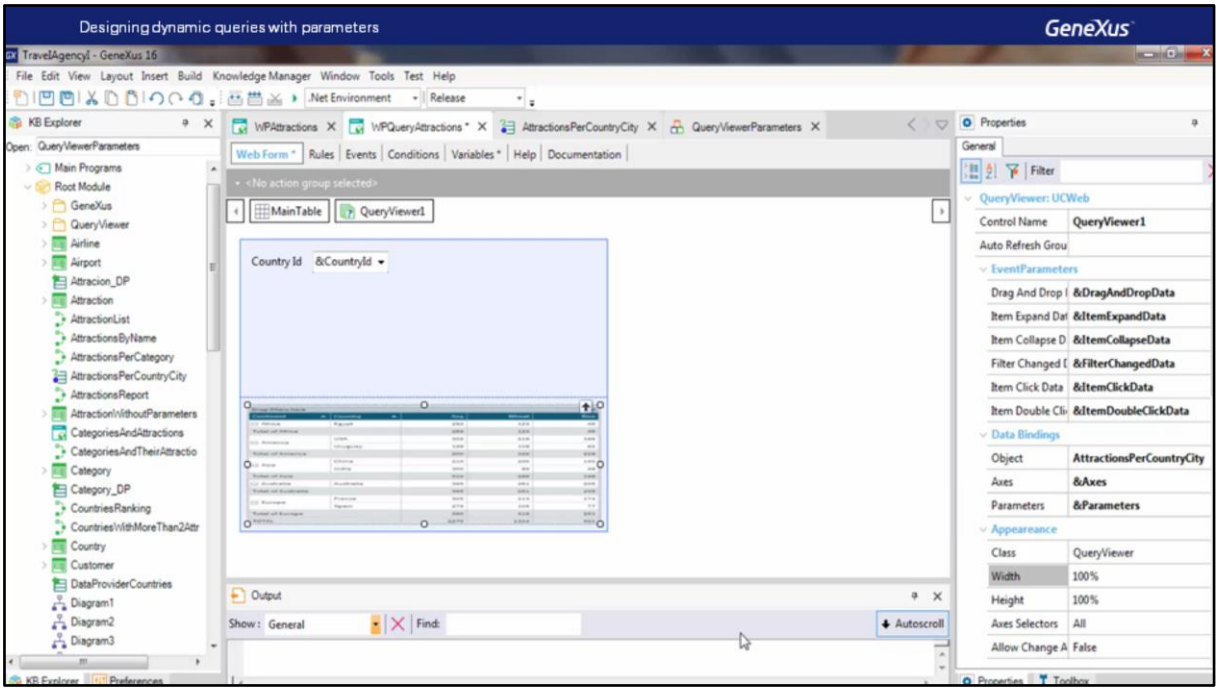
Let’s now see some further examples.



Suppose that now the travel agency requires to make some changes to the Query object that has been defined because they now want to have the possibility of querying the graph to find the number of tourist attractions located in the cities of a given country that is not necessarily France or China.
We will then need to make some changes in the definition of this Query object so that it receives the country as parameter.

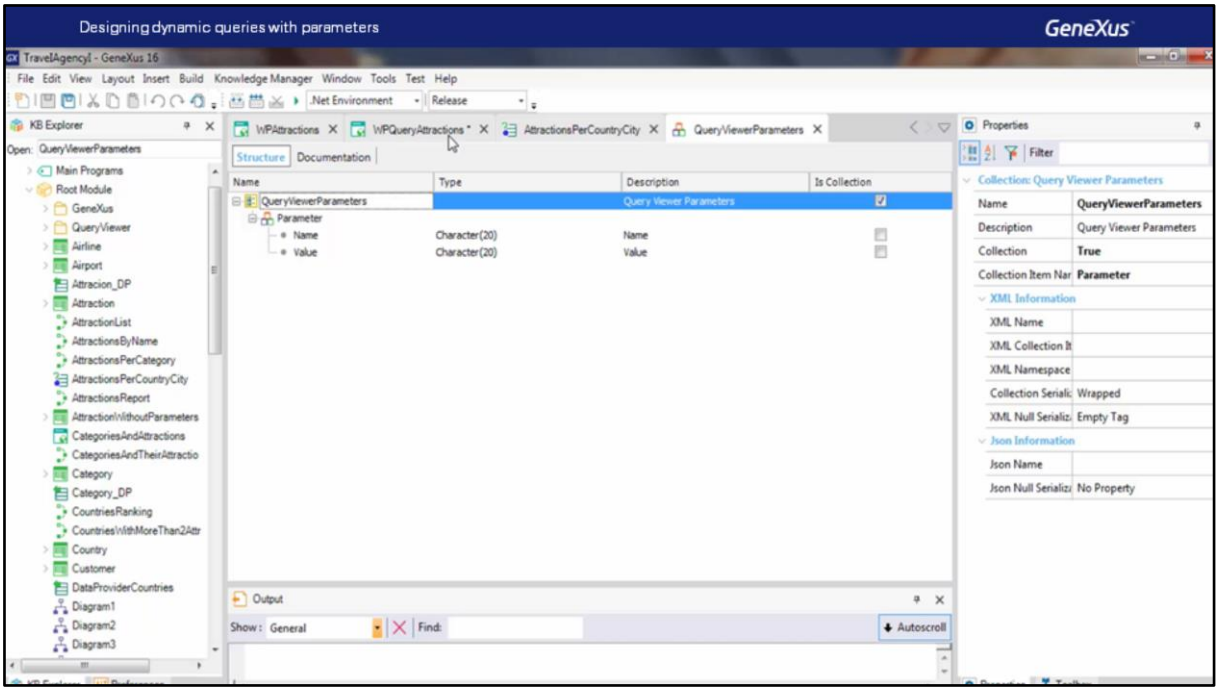
We will start by deleting the filter and the order defined.
At the parameters node we click enter, and there we define CountryId.
We set Country as description, and we also define the corresponding filter so that it will consider the value of this country in the query.

We save these changes and then edit the web panel to enable the user to select the country from a list, with that value being considered as parameter for the query.



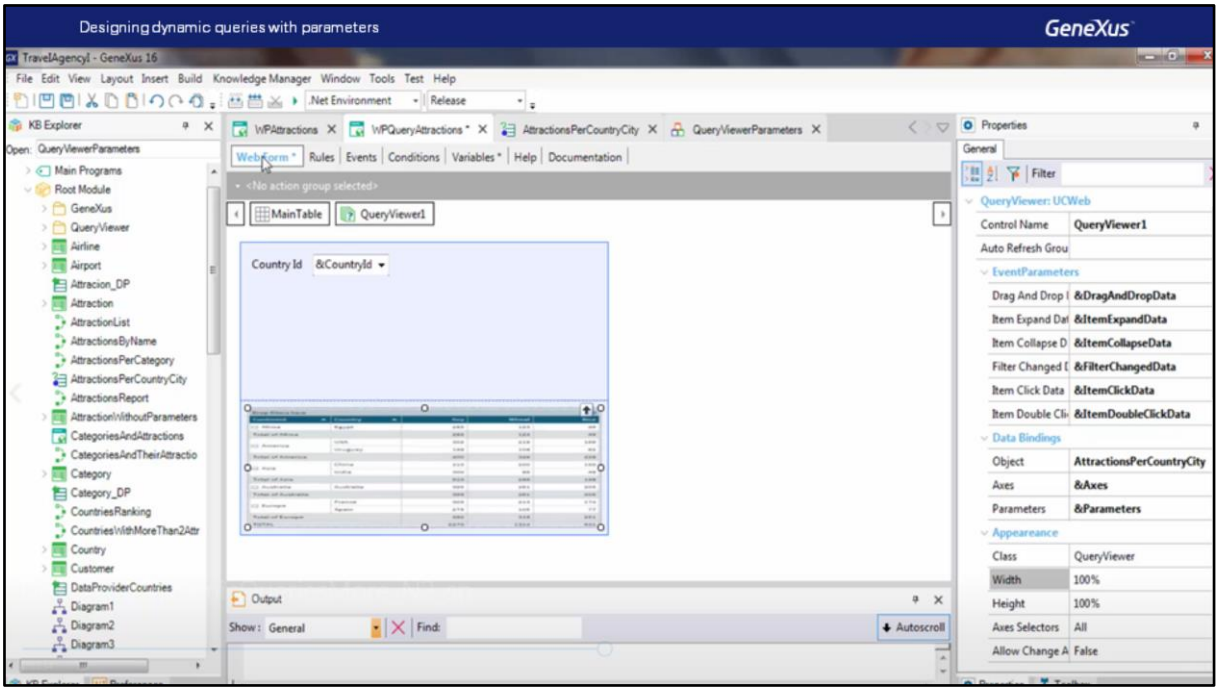
So, the first thing we do in the web panel is to define the &CountryId variable and define it as a dynamic combo so that it saves CountryId and value and shows CountryName as description.

In order to do this. we will edit the Control type property and select the Dynamic combo box value. There's indication that the Item value, that is, the value that will be saved, will be CountryId, and the value to be shown will be CountryName.
Now, what's still pending is for the QueryViewer control to consider the &CountryId variable as parameter.
Note that the Parameters property of the QueryViewer control has the &Parameters variable automatically associated. That variable is based on a structured data type that is automatically created upon dragging the QueryViewer control over the form. The variable was also defined automatically.



It we look at its structure, we will see that it's a collection of parameters, each with a name and a value. Something important to consider is that both items are of the Character type. This should be borne in mind at the time of loading this collection with the parameters to be considered, they must all be of the same data type.

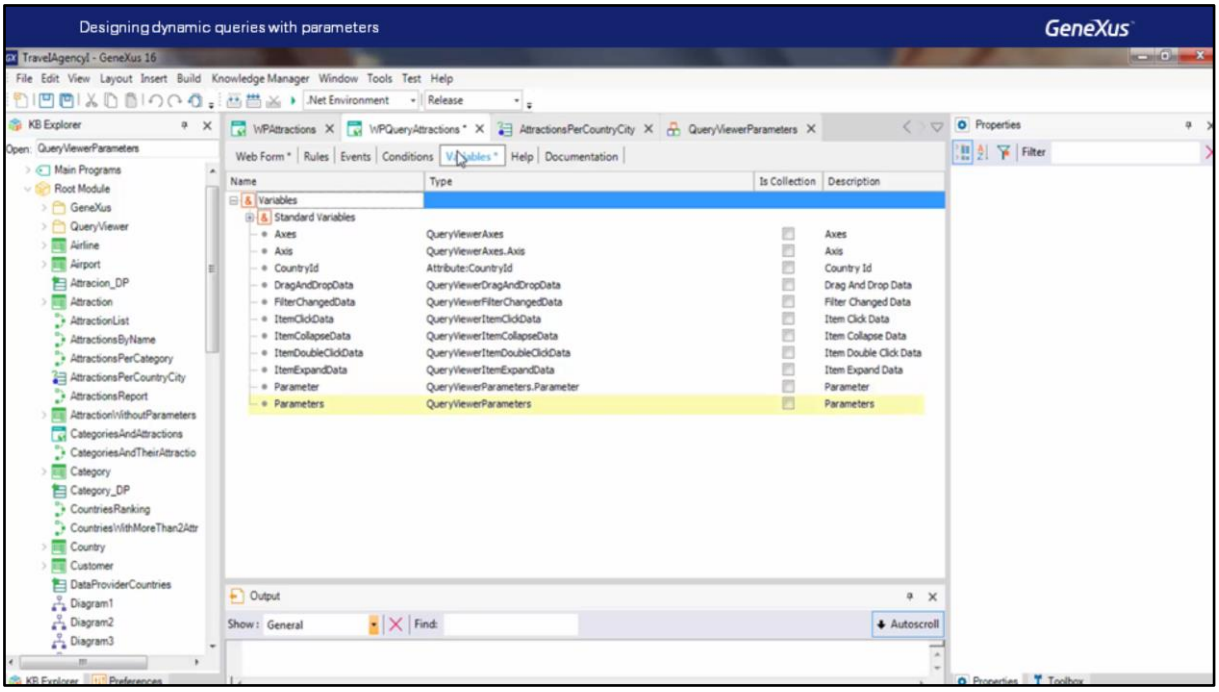
What we must do then, is to load the variable of our filter to the collection of parameters associated with the QueryViewer control.



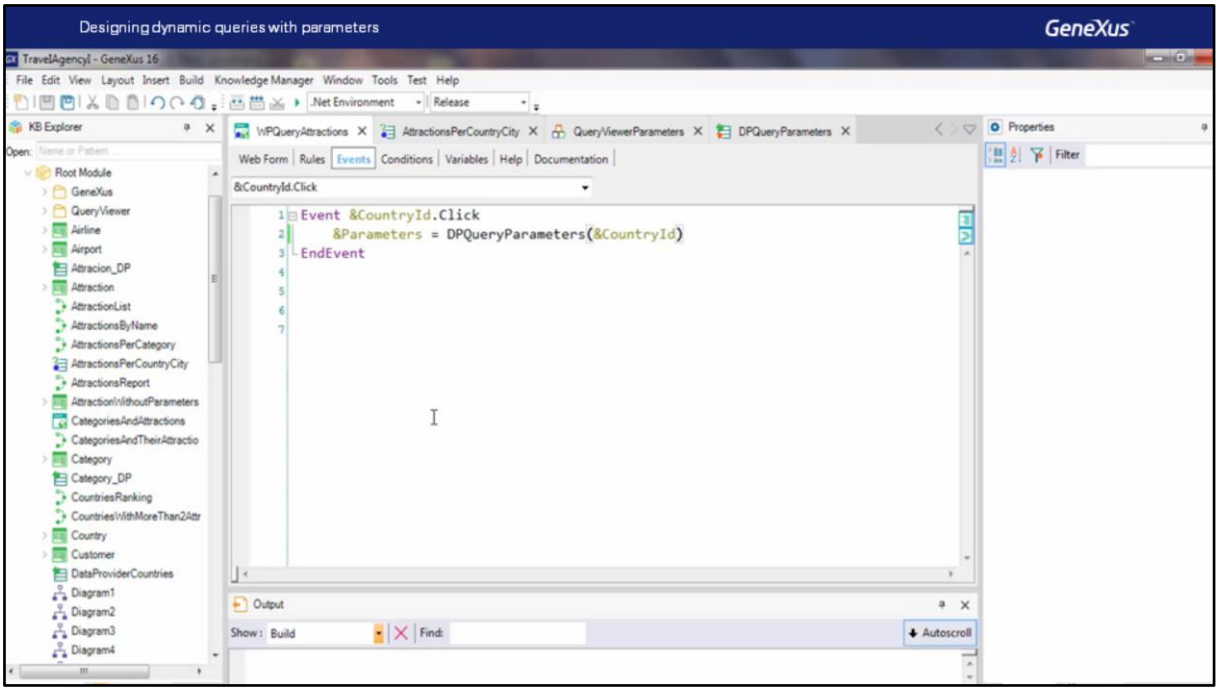
The question now is: Where will we code that load? In which event?
Let's consider the following:

What we want is for the parameter to be loaded when a country is selected from the list, thus updating the query to show the number of attractions in the cities of the country selected.

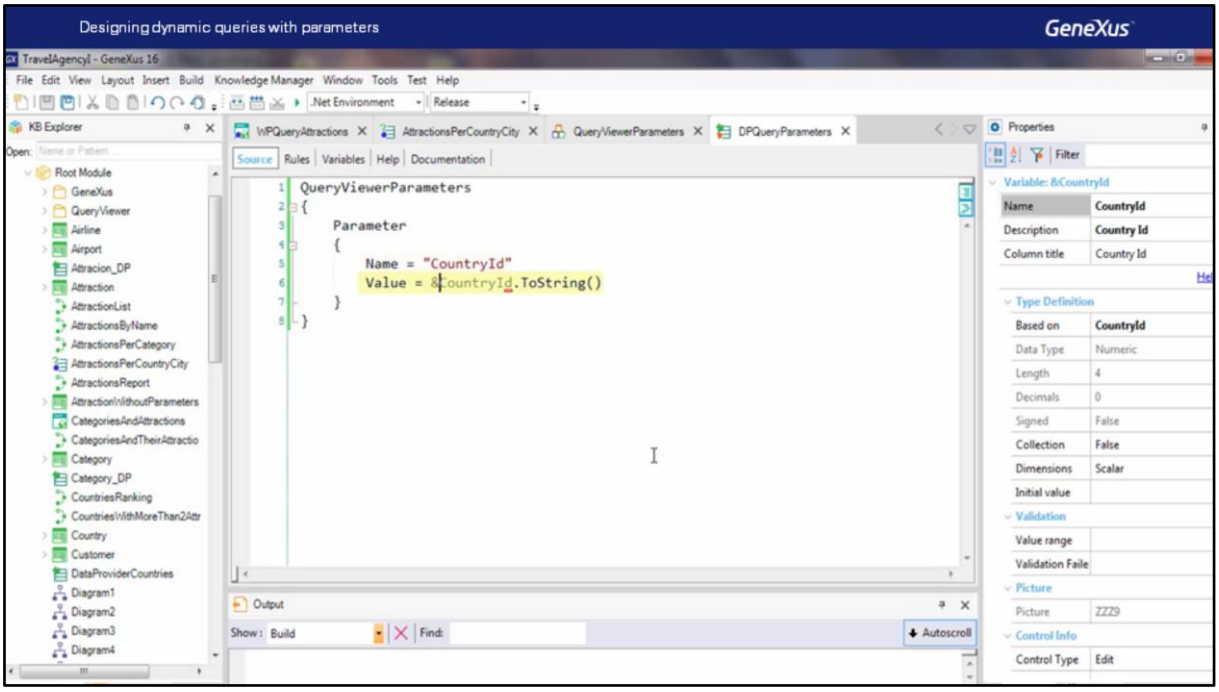
To achieve this we will code the Click event associated with the &CountryId variable. This event will be triggered every time that the user selects an element from the dynamic combo.



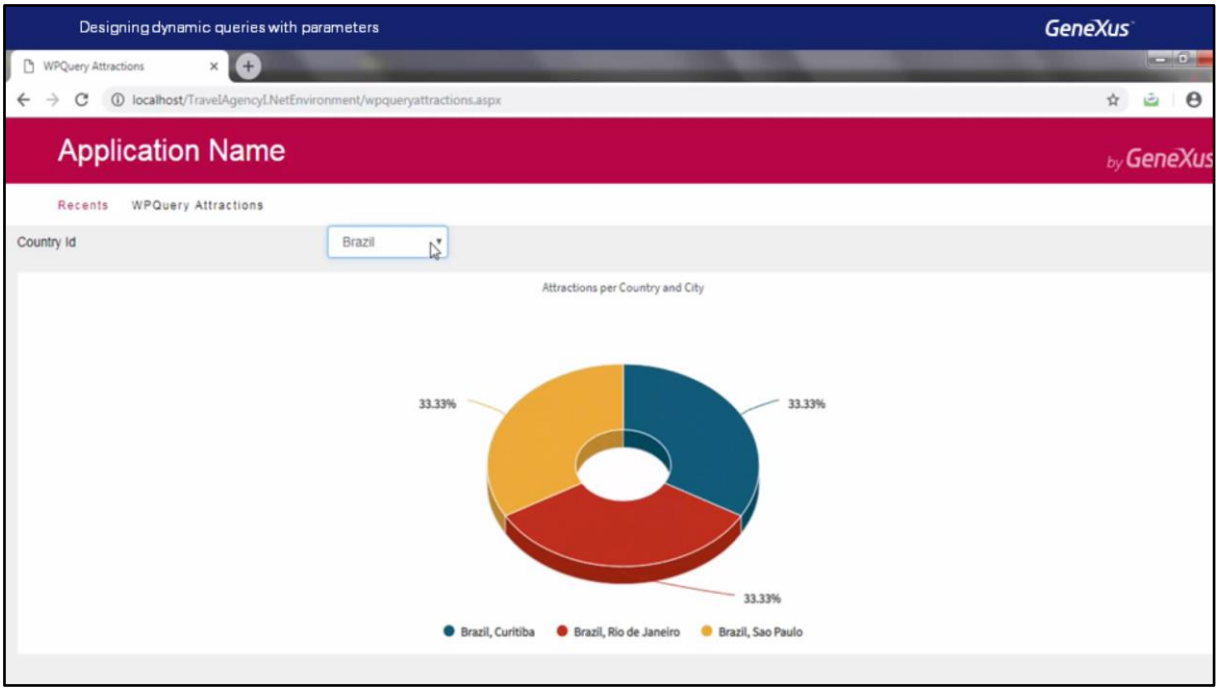
When we go to the variables sector we will see that the &Parameter variables have already been created, based on the data type that represents a parameter, and the &Parameters variable based on the data type that represents the collection of parameters.



Therefore, inside the events sector, and in the Click event associated with the &CountryId variable, we receive the load of the &Parameters variable through the output of a Data Provider.

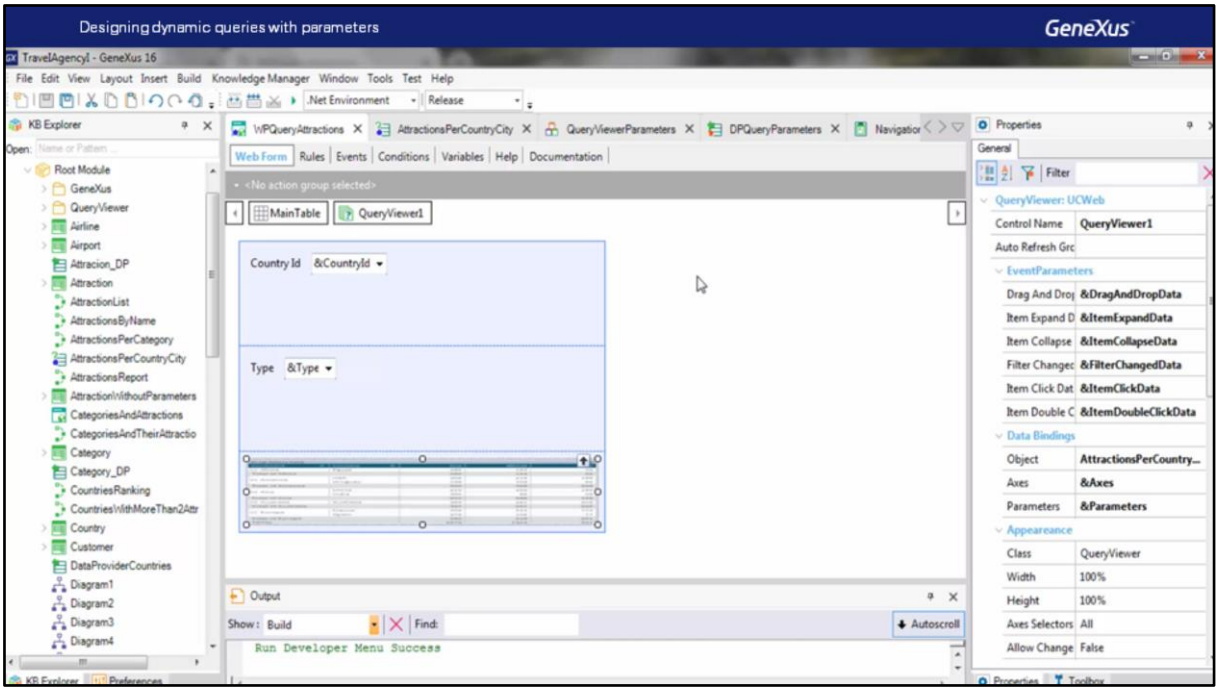


This Data Provider receives the value of the country selected and loads that parameter into the collection, indicating the name and the value considered. Since it is of the Character data type, we must convert it, and that's why we use the ToString method.



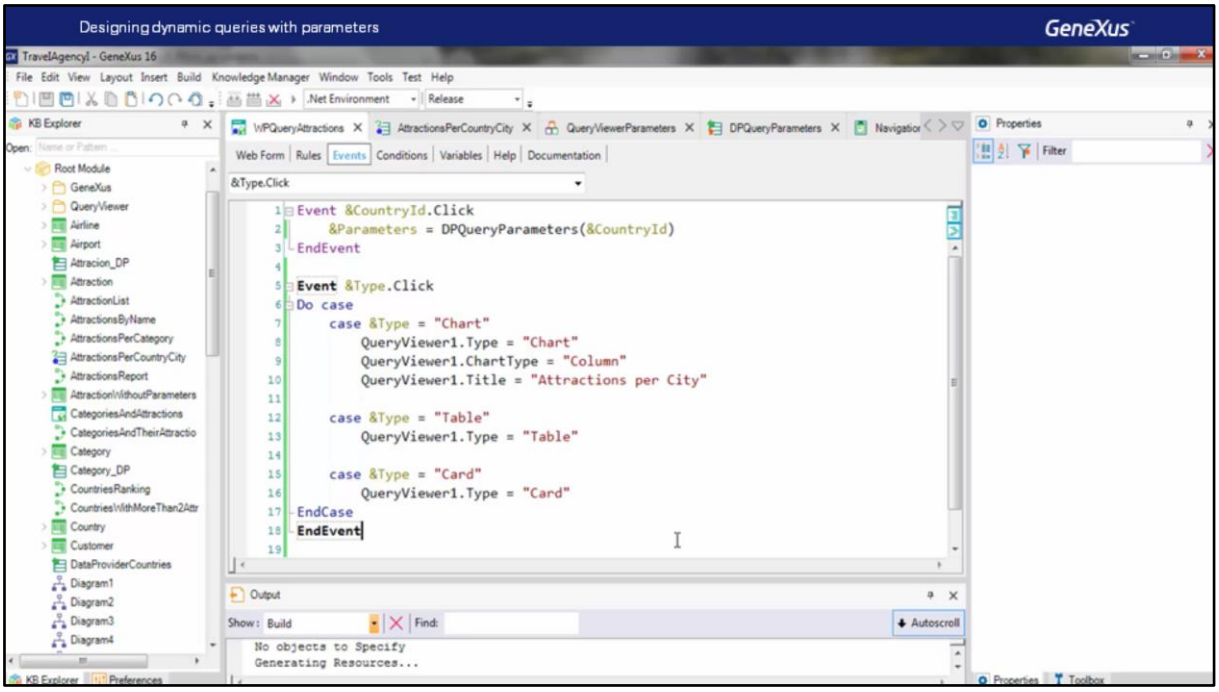
Let's go back to the web panel, and to view another type of graph in runtime, we will make a change by selecting Doughnout 3D.
And now, we press F5.

We select the web panel, and there we verify that, if we change the country, the query will be updated.



Now let's take a quick look at the possibility that we also have for selecting, in runtime, the type of output for the query. In our example, we will consider that the user wishes to have the possibility of deciding between viewing the output as a chart, as a table, or as a card.
So, we define a new &Type variable, of the Character type, that will appear as a combo with the Chart, Table, and Card options.

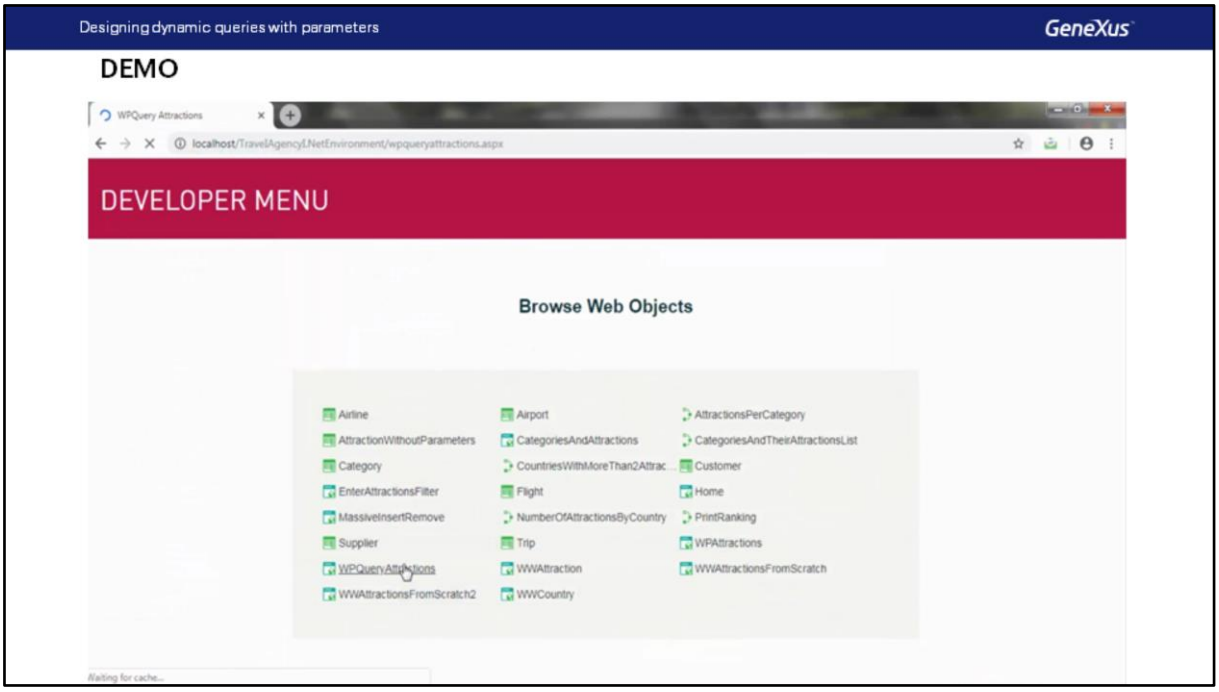
In the Control type property we select Combo box and assign the values.



We go to the events, and in the Click event associated with this variable, we will modify the value of the Type property of the QueryViewer control so that, for each case, it takes the corresponding value:
For that purpose, we use the control structure Do case, which allows us to consider what to do, depending on the value. In this case, it takes the variable &Type.
So, we indicate the name of the control, QueryViewer1, dot, the Type property, and then we indicate the value.

When the value selected is “Chart”, we will build a chart with columns, and will assign a title to it: “Attractions by city”.
When the type selected is “Table”, we will then view it as a table, or otherwise, as a card.

Let’s now press F5.



[DEMO: https://youtu.be/sf-H-k_JF2M]

Observemos que el formato Table nos permite, por cada columna, ordenar en forma ascendente, descendente, mostrar u ocultar valores.

E incluso, desde aquí, es posible exportar el resultado en distintos formatos.

GeneXus™

The power of doing.

More videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications