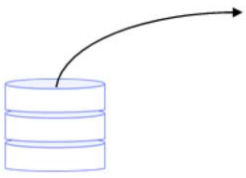
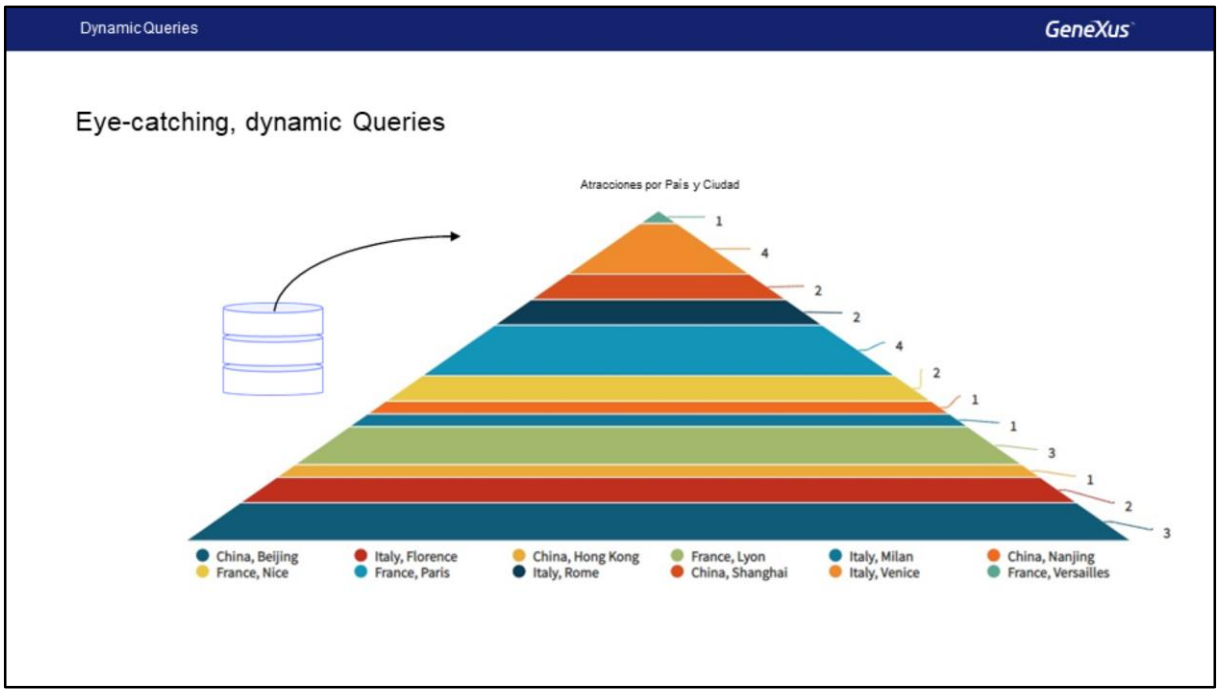
As we've said before, when designing enterprise applications we need to obtain the best reports and statistics that will help us make informed decisions.

Eye-catching, dynamic Queries



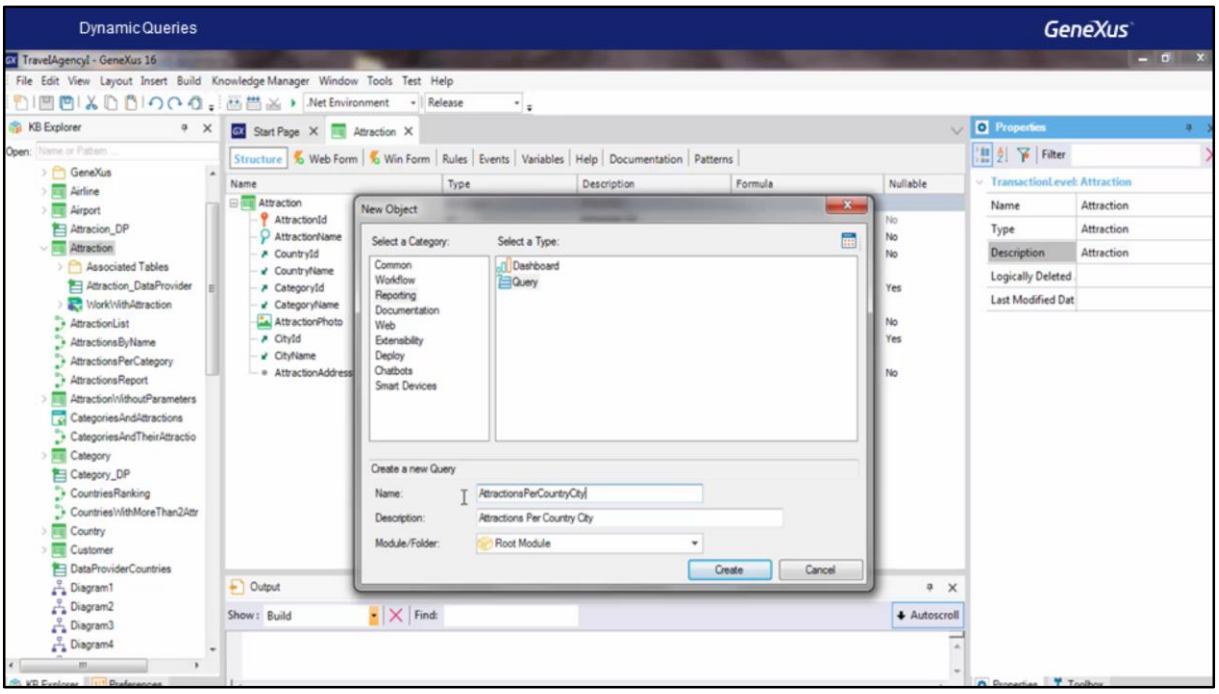
City Name	Category Name	Attraction Name	Quantity
Beijing	Famous Landmark	Great wall	1
	Total of Famous Landmark		1
	Monument	Dahilan	1
	Total of Monument		1
	Museum	National Museum	1
	Total of Museum		1
Total of Beijing			3
Curitiba	Famous Landmark	Barigui Park	1
		Botanical Garden	1
	Total of Famous Landmark		2
Total of Curitiba			2
Florence	Building	Palazzo Vecchio	1
		Santa Maria del Flor	1
	Total of Building		2
Total of Florence			2
Hong Kong	Monument	Hong Kong Henness	1
	Total of Monument		1
Total of Hong Kong			1
Lyon	Art Gallery	Galerie Art Club	1
		La Salle de Bains	1
	Total of Art Gallery		2
	Famous Landmark	Fourvielle Hill	1
	Total of Famous Landmark		1

The Query object allows creating eye-catching, dynamic queries in a simple and intuitive manner.



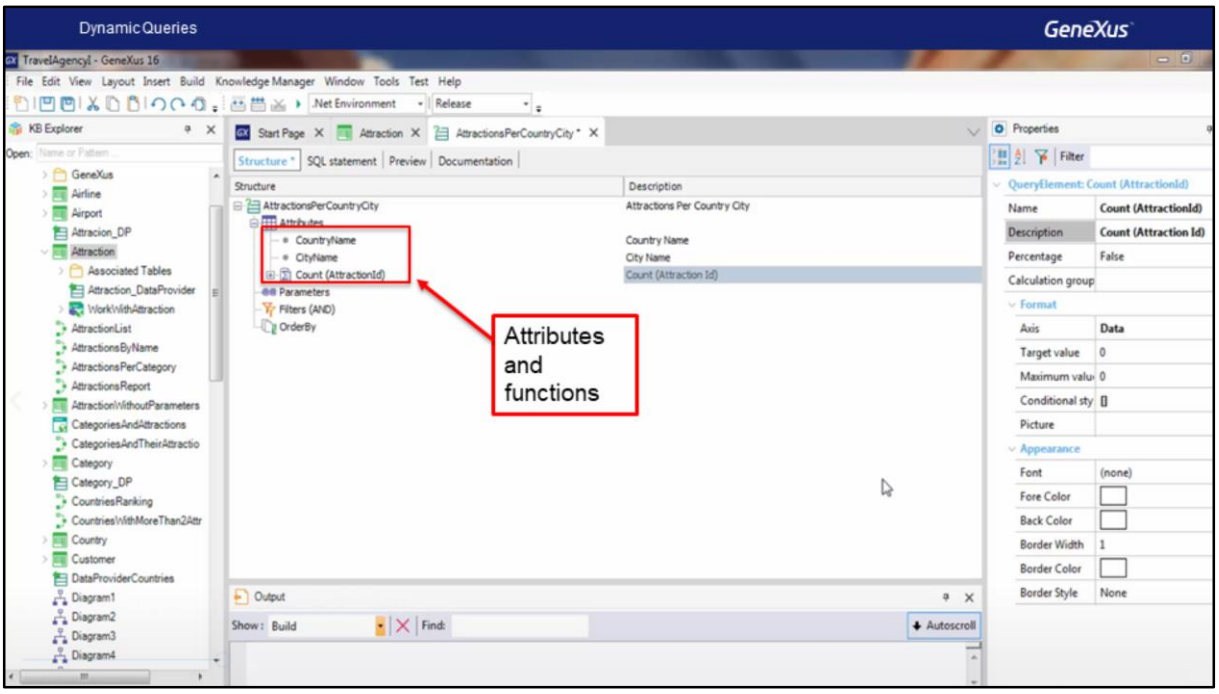
This increases and enhances the value of the information received from the database.

Let's see some examples:



The Travel Agency asks us to be able to chart the number of tourist attractions in each country and city.

So, we will create a new object; within the Reporting category we select the Query object type and call it AttractionsPerCountryCity.

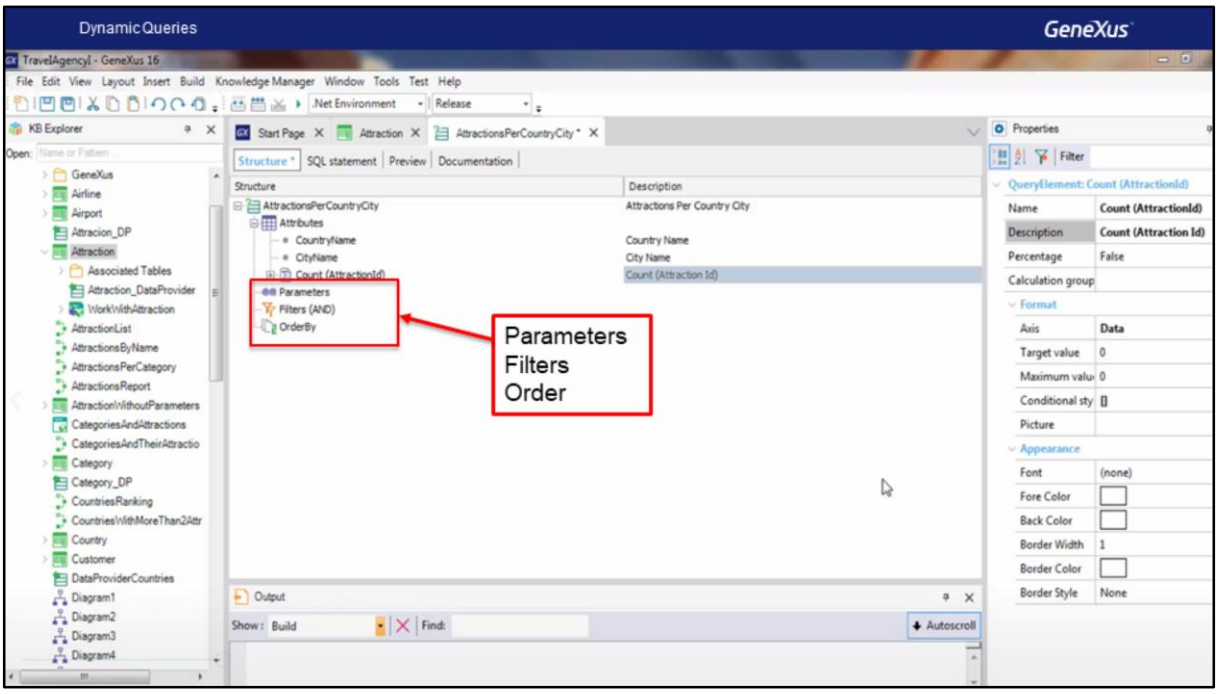


In this sector of the structure we will define the query. Let's analyze the meaning of the components displayed:
In the attributes node we must mention precisely all the attributes that will be included in the query. This statement is simple and made in flat list format.
But in addition to attributes here it is also possible to define functions, such as Count, Sum, Average, Maximum and Minimum, and it is also possible to include nested functions.
In this example, we want to chart the number of attractions per country and city, so we will add the attributes CountryName, CityName and state the formula Count(AttractionId).

If necessary, before applying a function, it is possible to group by some attribute, which allows viewing the data in different ways.
The syntax would be as follows:
Name of the function, attribute to which it is applied, and the set of attributes by which you want to define that grouping.

In this "Description" column is the tag that will be used to display the attributes in the result of the query. So, in the result of the formula count(AttractionId) we will enter the description "Attractions quantity."
If we also edit the value of this property, and select True, we can see the results of the formula as a percentage.

This option is available if we right-click and select Show as percentage.

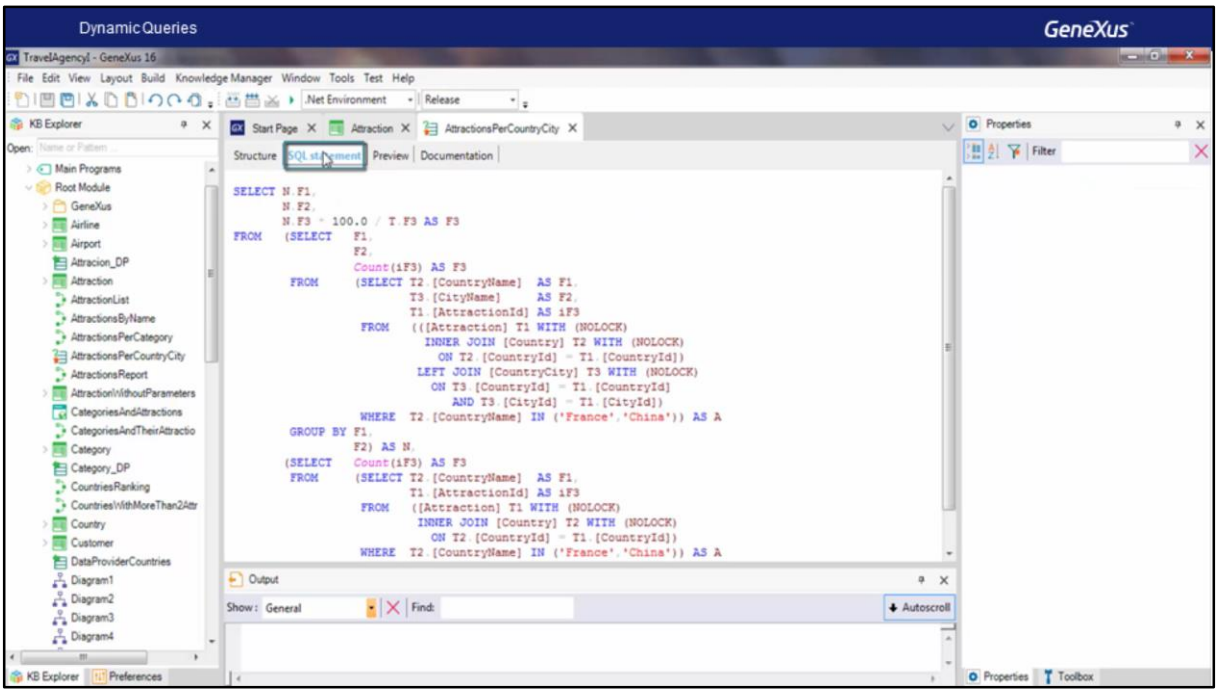


OK, let's move on to the parameters. The query can receive parameters, and it is here where we must declare them and set their properties.
The name, description and data type will be defined; also, if it is based on a domain or attribute, if it is a collection, and the predetermined value, if applicable.
Let's continue with the Filters.
The filters to be applied in the query will be specified here.
We can define a group of filters, which by default are linked by the connector AND... but we can also use the OR connector.
Let's quickly see some examples of the syntax used when declaring these filters or conditions:

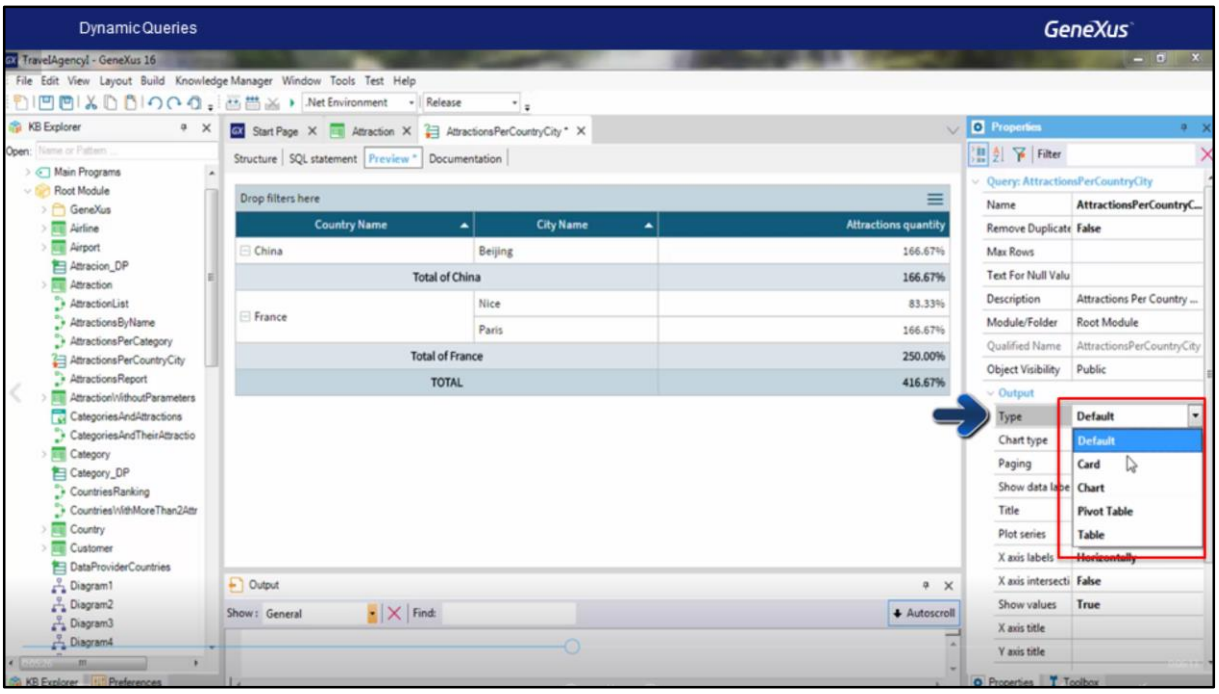
CountryId greater than or equal to three. This would be the filter to declare if we only want to see the number of attractions in the cities of the countries whose ID is greater than or equal to three.
Let's see another example: CountryId is a list of values to query, in this case, the countries with ID 1, 2 and 3.
Another example is CountryName, which is a list of values to query, for example, the cities of France, Italy, China and the United States.
In this example, we will define the query for the cities of France and China.
So, we type.
Lastly, in the Order section we can indicate the desired order to view the result of the query.
In this case, we will configure the query to be ordered by CountryName, so we state this order:

Lastly, in the Order section we can indicate the desired order to view the result of the query.
In this case, we will configure the query to be ordered by CountryName, so we state this order:

Well, we have completed the definition of our query so we save the changes.

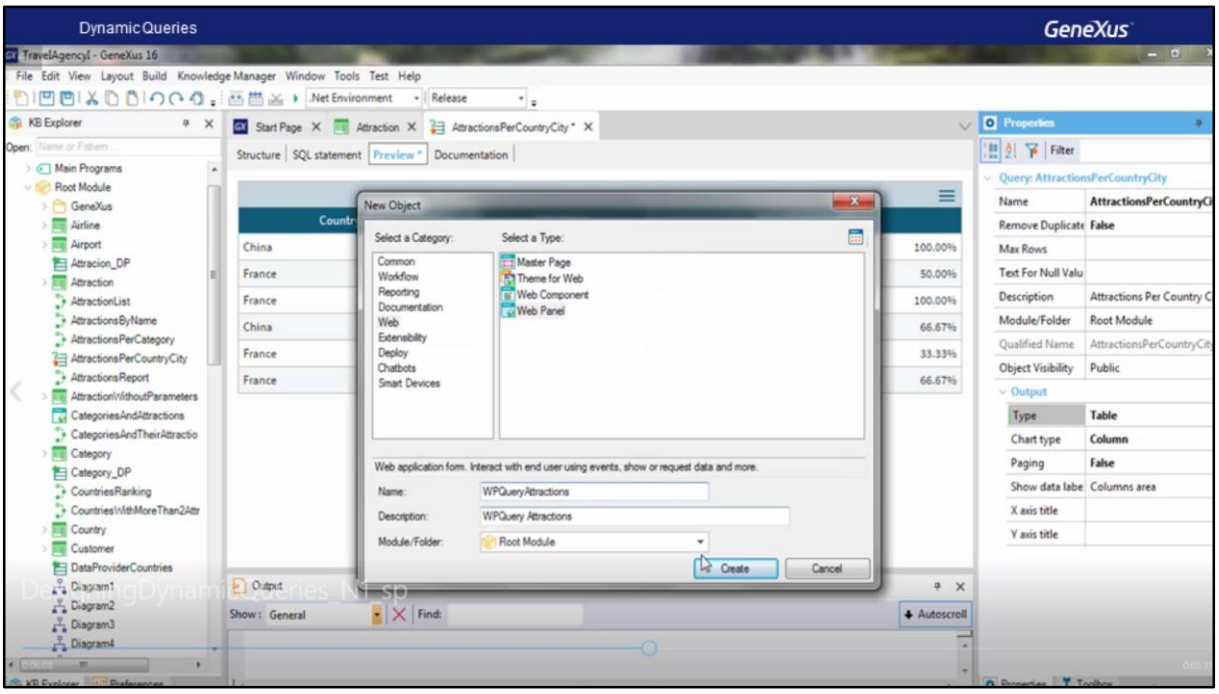


Look at this SQL statement tab. Here we can see the SQL statements that will be used to solve the query we have created.



The format of this preview can be changed from the Query object properties.
In this Type property we can select:
Card
Chart
Pivot table
Table

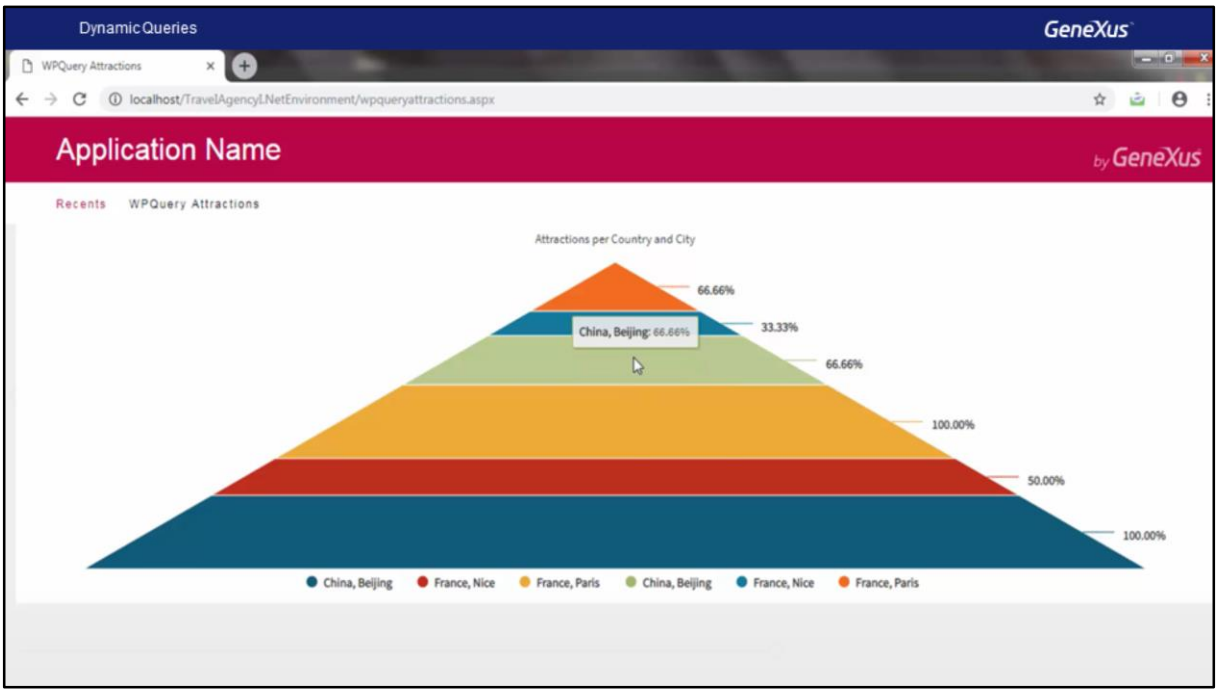
At this point, we have created our query and we have seen the previews available for the result. To actually view this query at runtime, it will be necessary to define a web panel and use the Query Viewer control that will be in charge of showing the result of the query in some of the ways we have tried.



So, we will create this web panel, and from the Toolbox we will drag the QueryViewer control.

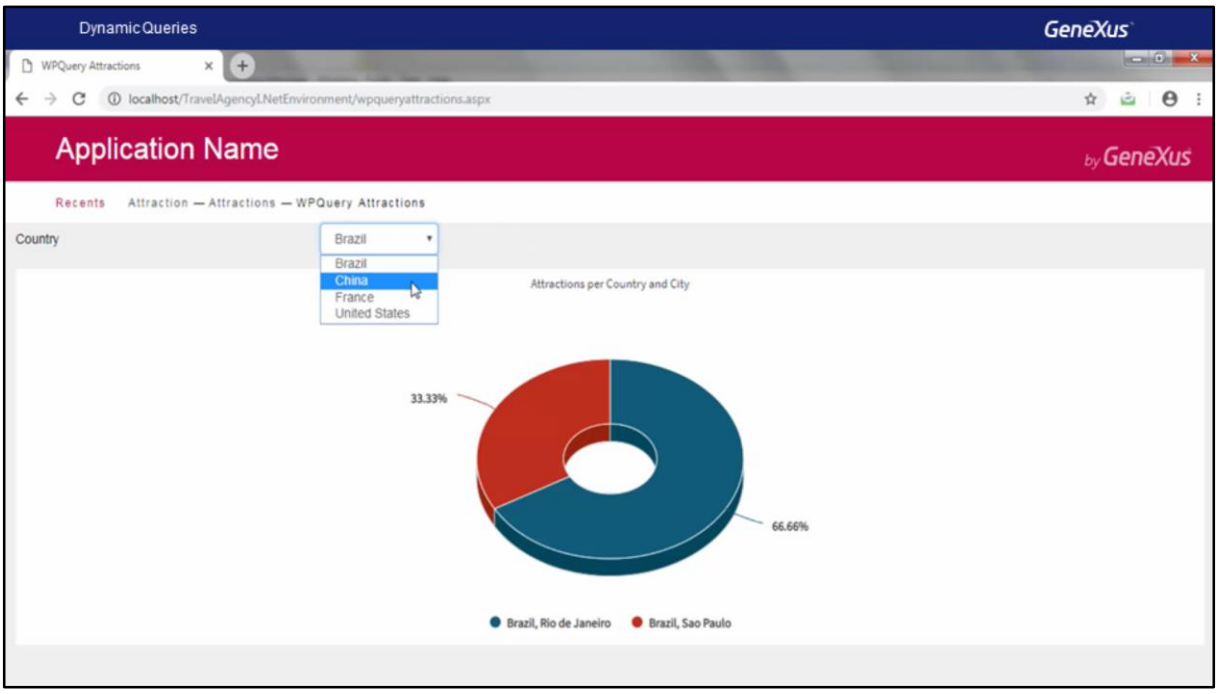
We edit the properties of this control, and in the Object property we will enter the name of the Query object to display. In this case, AttractionsPerCountryCity.

Then, to define the output, we need to configure the properties below the Output group.
So, in the Type property we select the Chart option. Next, we can choose the type of chart.
We'll choose Pyramid chart.
And enter a name for it: Attractions per Country and City.



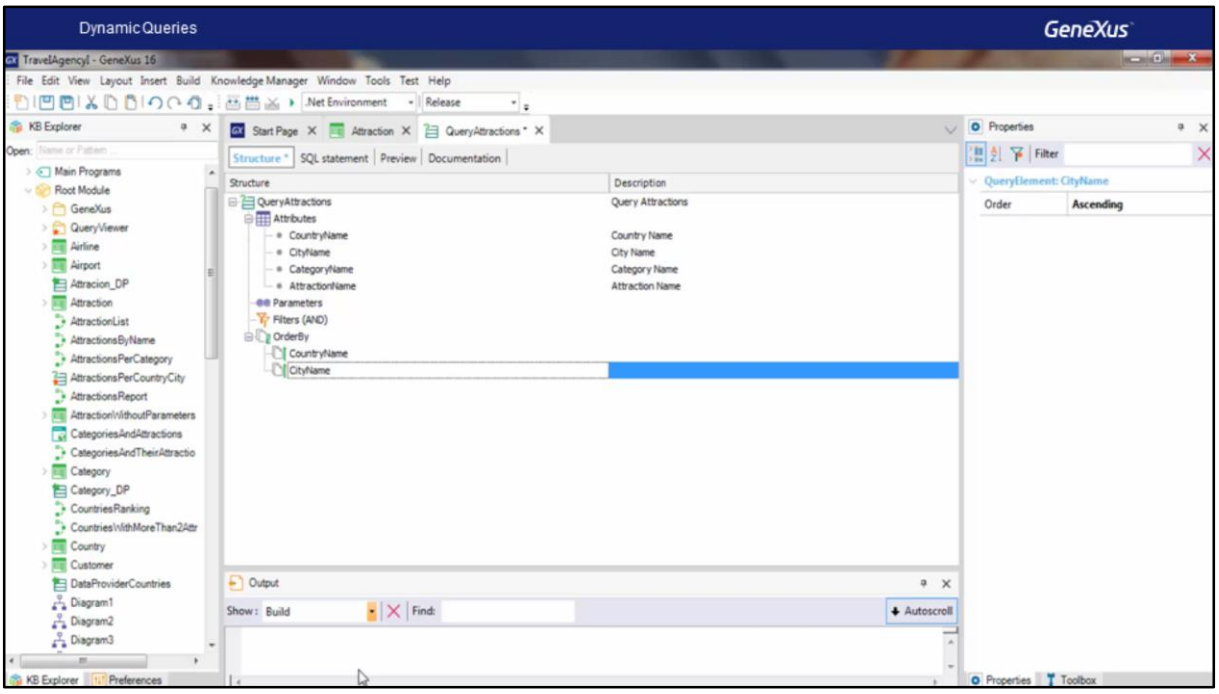
We can already run our app, so we press F5.

We select the web panel and see the chart at runtime.



Although we are not going to implement it in this video, it could happen that the Agency requests some modifications because they want to be able to view in the chart the number of tourist attractions in the cities of a certain country that doesn't necessarily have to be France or China, but some country selected by the end user.

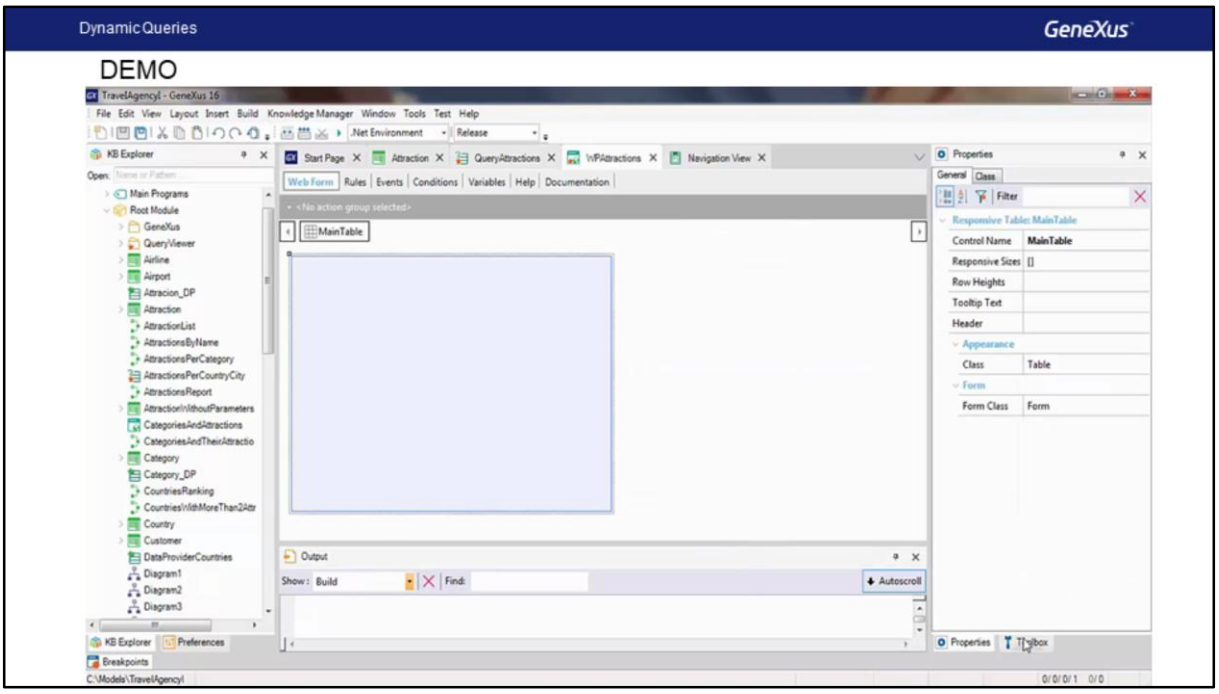
To meet this requirement we would have to configure the Query object to receive CountryId as a parameter and achieve this behavior at runtime:



Let's see another example. The travel agency wants to be able to see in a pivot table all the tourist attractions of each country and city grouped by category.
To this end, we create a new Query object and call it QueryAttractions.
As we said, we want to see the tourist attractions grouped by category, for each city in each country, so we must add the attributes we want to view in the order in which we want to define that grouping.
So, first we type:
CountryName
CityName
CategoryName
AttractionName

Also, the query should be ordered by Country and City name, so in the Order sector we enter CountryName, CityName.
We save this query...

And create a new web panel.



[DEMO: https://youtu.be/6xTb7s5k9Sc]

From the Toolbox, we drag the QueryViewer control, and in its Object property we associate the Query object we've just created.

To configure the output format, we open the Type property and choose Pivot Table.
And press F5.

One of the characteristics of this output type is that we can change the place of the columns, for example, and the information will be regrouped accordingly.
For example, we switch the columns CityName and CategoryName...

We see that the information has been regrouped according to this new order.
Let's return to the original order.
Note also that for each column we can sort the information in ascending or descending order, and show or hide different data.
For example, we will sort the column CategoryName in descending order and remove the subtotals.

Also, note that from here we can export the query to different formats.

For example, we will export it to Excel.

Lastly, we update the changes in GeneXus Server.

GeneXus™

The power of doing.

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