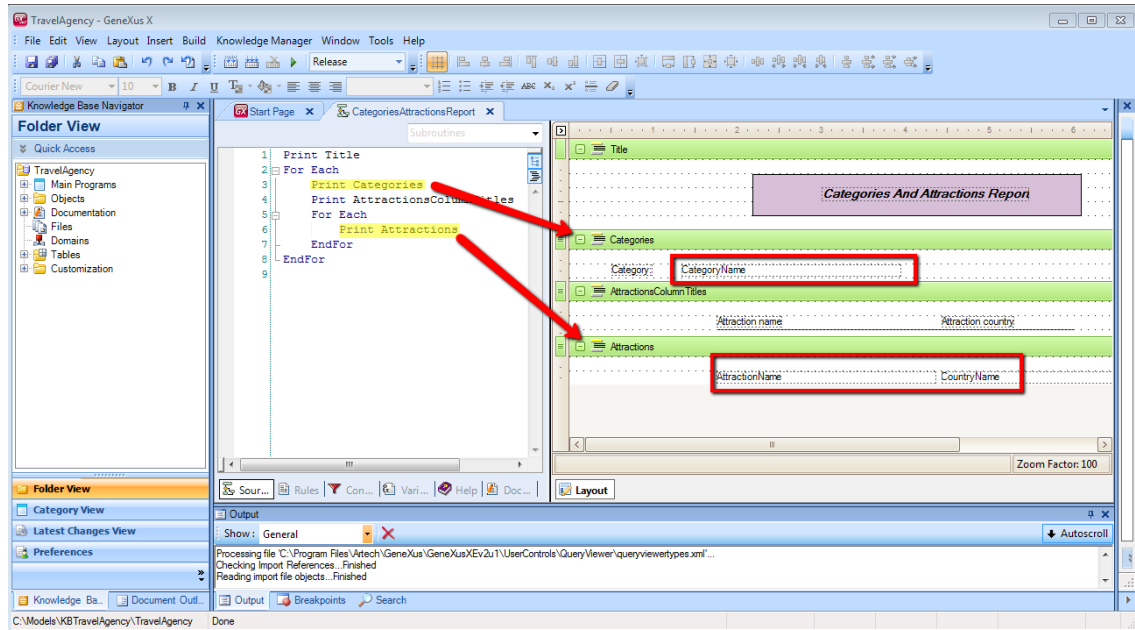
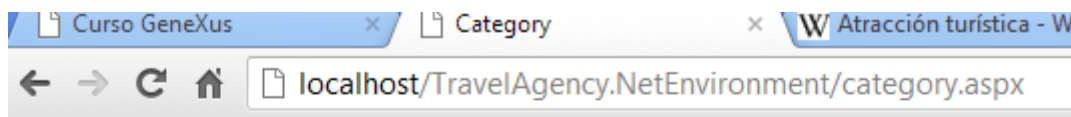


## GeneXus course - Nested For Each command to list grouped data

We already implemented a procedure that listed the **categories** or tourist attractions, and the **attractions** in each of them.



What would have happened if the categories we had in the database did not have any related tourist attractions? For example, let's enter a new category such as Art Gallery.



## Application Header

Recents: [Attraction](#) [Work With Attractions](#) [Category](#)

Category

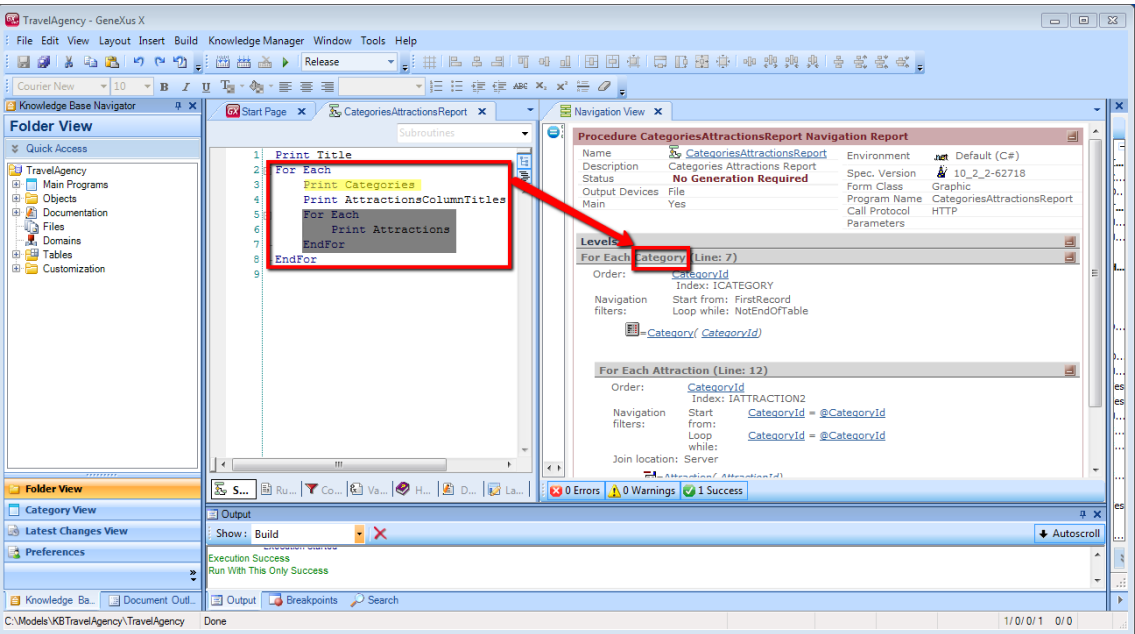
Id

Name

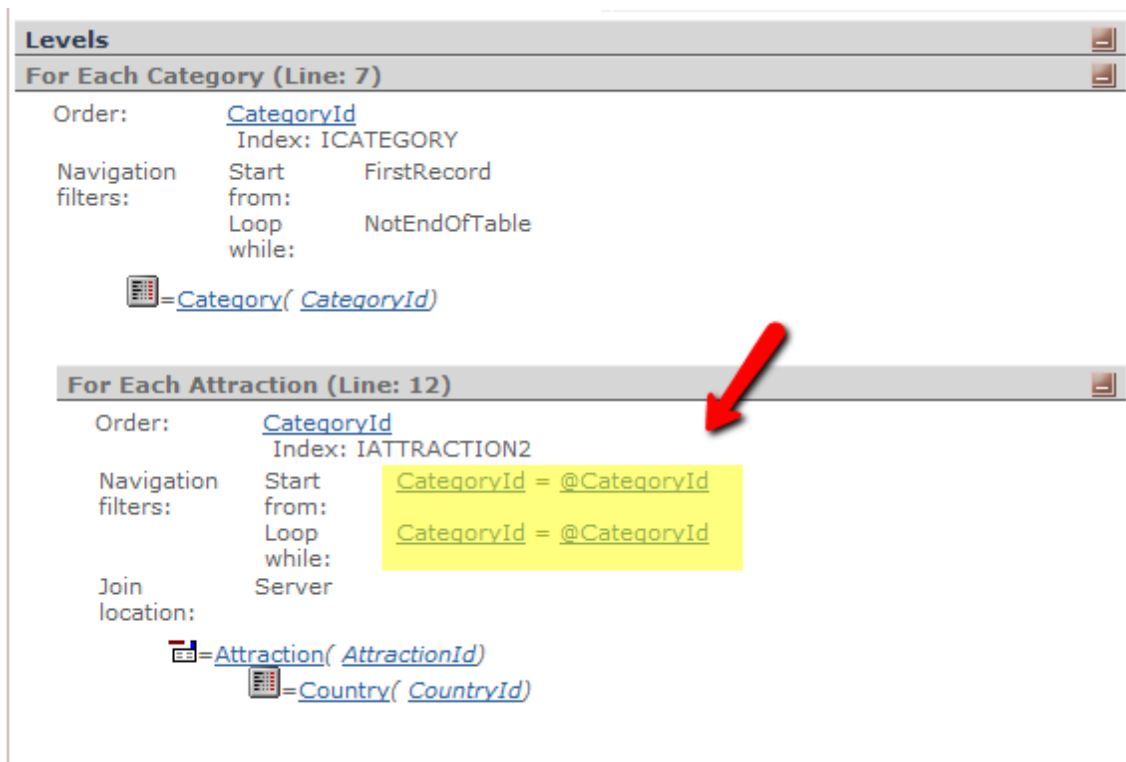
It still does not have any tourist attraction associated with it. If we now execute the listing, we will see that the category appears without any attractions.

Categories And Attractions Report	
Category: Museum	
Attraction name	Attraction country
Louvre Museum	France
The Smithsonian Institute	United States
Category: Monument	
Attraction name	Attraction country
Eiffel Tower	France
The Christ Redeemer	Brazil
Category: Famous Landmarks	
Attraction name	Attraction country
Great Wall	China
Egypt Pyramids	Egypt
Category: Art Gallery	
Attraction name	Attraction country

This is because the external For each goes over the categories table regardless of whether there are related records in the attractions table or not.



Note that for the recently entered Art Gallery category, the data and titles will be printed, and when the nested For each is accessed afterwards, none of the records will fulfill this condition:



Because there are no attractions for that category, yet!

-----

But, what if we wanted only for categories **with** attractions to be listed?

We have an efficient way to do this by minimizing the number of times that tables have to be gone over (if this is done too many times, then we would be wasting valuable time on it).

This may be achieved by going over the attractions table only **once**. How?

By grouping per category, and for each group.

We must find the name of the category in the Categories table, and then print it along with the titles...

Then we print the name of the attraction and country for each attraction in the group.

When we change groups, we search for the name of the current category again, Monument, and then print it along with the titles. And, again, for each attraction in the group we print its name and country.

We would do the same for the last group, corresponding to Famous Landmarks.

We should note that in the categories table, the number 4, Art Gallery, will not be on the listing because we do not **go over** the Category table but only the Attraction table (we only access Category to search for the name).

So, how can we modify the For each we had to achieve this change in behavior?

First we will need to make Attraction (instead of Category) the base table of the external For each, just as it is the base table of the nested For each. That way, only **that** table will be gone over.

Then we will have to instruct it to group by category as it goes over Attraction. This is possible with the Order clause.

When a pair of nested For eachs share the same base table, the attributes included in the order clause of the external For each will be the attributes by which the information will be **grouped**.

So, how can we make Attraction become the base table of the external For each, when the only attributes we are using in that For each are CategoryId and CategoryName (the latter originating in the Categories print block)?

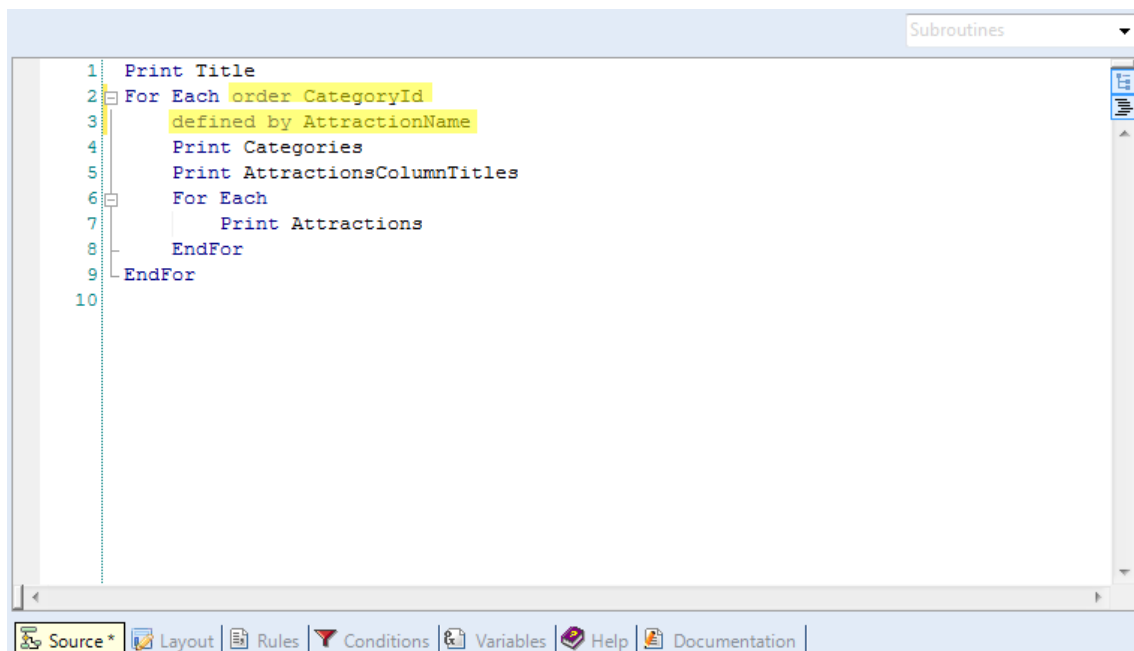
It will be enough with including, inside the For each, an attribute from the Attraction table, such as AttractionName. To do this we have the Defined by clause of the For each.

This clause allows us to name attributes, just so they take part, **along with the other ones in the For each**, in determining the base table.

We can try all this in GeneXus...

-----

In the external For each we add the order clause by CategoryId and the Defined by clause,



```
1 Print Title
2 For Each order CategoryId
3   defined by AttractionName
4   Print Categories
5   Print AttractionsColumnTitles
6   For Each
7     Print Attractions
8   EndFor
9 EndFor
10
```

The screenshot shows a code editor window with a 'Subroutines' dropdown at the top right. The code is as follows:

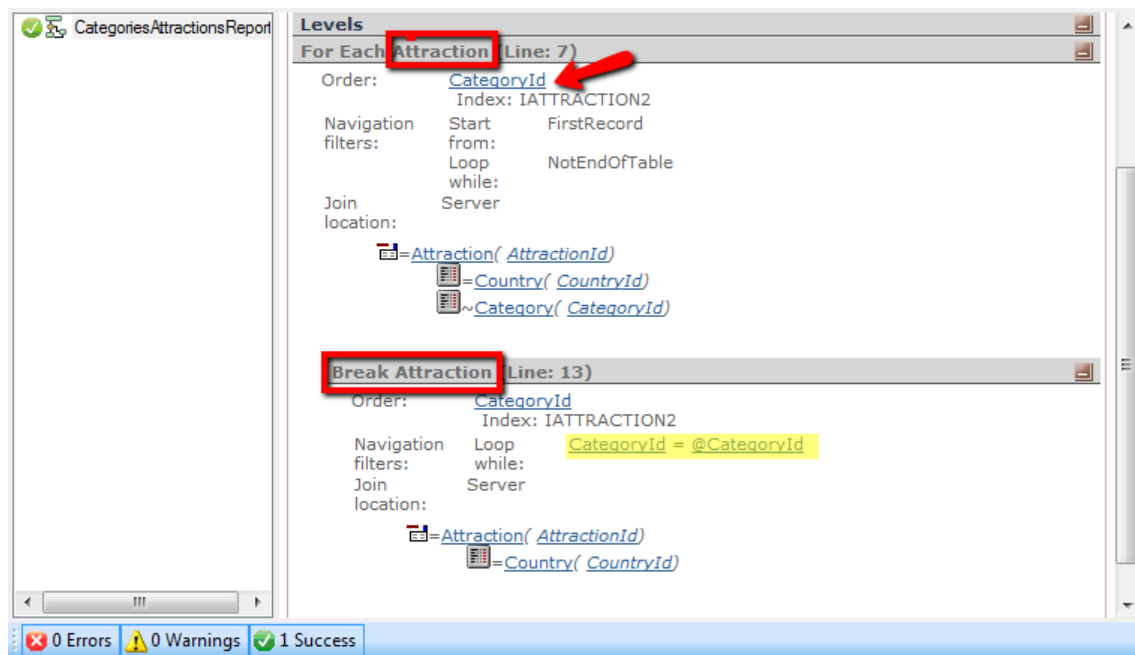
```
1 Print Title
2 For Each order CategoryId
3   defined by AttractionName
4   Print Categories
5   Print AttractionsColumnTitles
6   For Each
7     Print Attractions
8   EndFor
9 EndFor
10
```

The editor has a toolbar at the bottom with icons for Source, Layout, Rules, Conditions, Variables, Help, and Documentation.

and when we execute the procedure...

we will see that the format of the information we show is the same, but the Art Gallery, as we wanted, is not shown.

If we take a look at the navigation listing...



We can see that, the base table of the external For each was in fact changed to Attraction, and now, in the navigation of the nested, we see the word Break instead of Foreach, and immediately after, the Attraction base table, identical to that of the first For each.

Whenever we have two (or more) For eachs, one inside the other, going over the same table, we will have this case of grouping, known as **control break**, where the same table is gone over, in groups. The way to indicate the attribute by which we wish to group is to mention it in the order clause of the external For each.