Web Panels

How to Implement Control Breaks in Nested Grids

GeneXus

Web Panel with SEVERAL Grids

In another video we learned how the base tables and the navigation of a web panel with several grids were determined.

GeneXus

Web Form Rules Events Conditions Variables	Attraction Attractionld AttractionName AttractionPhoto Cityld AttractionPhoto Cityld AttractionAddress CountryCity CountryVid Cityld Cityld Cityld Cityld Cityld Cityld Cityld Citylo Ci
Country Name CountryName	Country Name &CountryName Category Category Category Category CategoryName GRID
City Name CityName GRID Attraction Id Attraction Name AttractionName Trips &trips &update2 &newTrip Total Trips &totalTrips	City Name &cityName GRID Attraction Id Attraction Name & Trips & Bupdate2 & &newTrip Total Trips &ctotalTrips
Total Attractions &totalAttractions	Total Attractions & total Attractions

In particular, we saw an example of nested grids performing a join.

That is, the external grid ran through a table with a 1 to N relationship with the table run through by the internal grid, regardless of whether these grids were implemented with or without a base table.

In both solutions, the web panel received a country identifier in a parameter, and the external grid showed the cities of that country; that is to say, it was going to run through CountryCity; and the internal grid showed the tourist attractions of that city. In other words, it was going to run through Attraction.

in several Grius. Hesteu	Event Grid1.Refresh PSeudocode &totalAttractions = 0 endevent
Web Form Rules Events Conditions Variables	For each Country.Clty
1 □ parm(in: CountryId);	where CountryId = @CountryId
Country Name CountryName	Event Grid1.Load &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions endevent
GRID	Load
City Name CityName	Event Grid2.Refresh <u>&totalTrips</u> = 0 Endevent
Attraction Id Attraction Name AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName Image: AttractionId AttractionName	For each Attraction order AttractionName where CountryId = @CountryId where CityId = @CityId
Total Trips &totalTrips Total Attractions &totalAttractions	Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent
	Load
Grid1 and Grid2 with Base Tables	endfor endfor

In the case of grids with a base table, this join was established automatically, without having to do anything. GeneXus detected it and added the filter in its source program.

	Event Grid1.Refresh PSeUdOCOde &totalAttractions = 0 endevent
Web Form Rules Events Conditions Variables	Event Grid1.Load For each Country.City &CountryName = CountryName &cityName = CityName &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions
Country Name &CountryName	Load
GRID	endfor endevent
City Name &cityName GRID	Event Grid2.Refresh &totalTrips = 0 Endevent
Attraction Id Attraction Name Trips &AttractionId &AttractionName &Etrips Total Trips &totalTrips Total Attractions &totalAttractions	Event Grid2.Load For each Attraction order AttractionName where CityName = &cityName &AttractionId = AttractionId &AttractionName = AttractionName &AttractionPhoto = AttractionPhoto &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Load endfor
	Endevent
Grid1 and Grid2 without Base Tables	endfor

On the other hand, in the case of grids without a base table, we had to specify it in the For each that we implemented to load the nested grid (and we only filtered by city, because the filter by country was already implicit by receiving it in a parameter in the Countryld attribute).

S Countries	× 🕄 View	Country Info_related Copy1 ×	+			0	- 0 ×
$\leftarrow \ \ \rightarrow \ \ \mathbf{C} \mathbf{\dot{C}}$	trialapps3.genexus.com	m/ld111ff2ece4700f8312f2a9	8d30e5285c/viewcountryinfo_rel	latedcopy1.aspx?CountryId=2		☆ 🛆	* 🗊 🛞 :
Recents	WWAttractions From	. — Countries — View Cou	ntry Info				· · · · · · · · · · · · · · · · · · ·
COUNTRY NAME	France						
	City Name	Paris					
	Attraction Name				Trips		
	Elffel Tower		Å		5	UPDATE	NEW TRIP
	Louvre Museum		as-at-		1	UPDATE	NEW TRIP
	Total Trips	La	6				
	City Name	Nice					
	Attraction Name				Trips		
	Matisse Museum				2	UPDATE	NEW TRIP
	Total Trips		2				
Total Attractions		3					

Here is the web panel with both grids **with a base table**. We have added an action in the countries Work With pattern to invoke this web panel.

If we choose France: here we see the attractions of Paris and Nice, which are the two cities we have entered in the system.

S View Country Info	_related Copy1 × 🔇 View	Country Info_related Copy1 ×	+				0	- 1		
\leftrightarrow \rightarrow G \heartsuit	trialapps3.genexus.co	m/ld111ff2ece4700f8312f2a	98d30e5285c/viewcountryinfo_relatedo	copy1.aspx?CountryId=1		☆	۵ :	* 2	: 🕲	
Trave	l Agency									
Recents	WWAttractions From	— France — Brazil — C	ountries — View Country Info							
COUNTRY NAME	Brazil									
	City Name	Rio de Janeiro								
	Attraction Name				Trips					
	Christ the Redemmer		The second		0	UPD	ATE	NEV	W TRIP	
	Total Trips		0							
	City Name	Sao Paulo								
	Attraction Name				Trips					
Total Attractions	Total Trips	1	0							

Now, let's see what happens if instead of choosing France we choose Brazil, for example, which also has two cities entered.

We see that for the first one, Rio de Janeiro, a tourist attraction is shown, but for the second one, Sao Paulo, no tourist attractions are shown. This is precisely because it is a **join**.

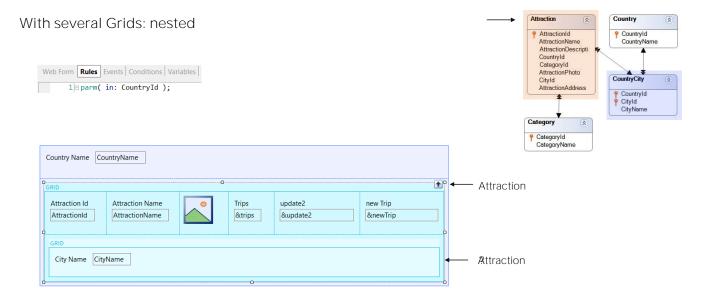
Web Form Rules Events Conditions Variables 1 parm(in: CountryId);	
n	Grid1.Refresh()
Country Name CountryName	Grid1 → Load
	Grid2.Refresh()
GRID City Name CityName	Grid2 \rightarrow Load Christ the Redeemer
GRID	Grid1 → Load()
Attraction Id Attraction Name Trips AttractionId AttractionName & trips & trips & trips & trips	Grid2.Refresh()
	Control Break instead of Join

The Refresh of the external grid will be executed first, and then, once positioned in the first city, Rio de Janeiro, it will be loaded in Grid1; right after that, the Refresh of the nested grid will be executed, and then the attractions of Rio de Janeiro will be loaded, which in this case is only one, Christ the Redeemer.

Then the next city, Sao Paulo, will be loaded and the nested grid will be refreshed. But when running through the table of attractions to load only those of Sao Paulo, none will be found.

In order to show only cities with attractions, we need to implement a **control break** and not a **join**.

GeneXus



We had outlined a case where an unwanted control break would occur instead of a join. But we had only addressed that by not specifying a base transaction for the nested grid. It appeared to be CountryCity, but actually it would be Attraction. We hadn't looked closely at the navigation, and now we will do so with our example.

	Rules Events Conditions Variables	Y AttractionName AttractionName AttractionDescripti CountryId CategoryId AttractionPhoto CityId AttractionAddress
P	Country Name &CountryName	Category
Country Nam	GRID	CategoryId CategoryName
GRID	City Name &cityName	
City Name	GRID	
GRID	Attraction Id Attraction Name Trips	For each Attraction or
Attraction I AttractionI	&AttractionId &AttractionName & & & & & & & & & & & & & & & & & & &	print CountryName print CityName
Total Trips	Total Trips &totalTrips	For each Attraction print Attraction
	Total Attractions &totalAttractions	endfor
Total Attractio		endfor

Attraction 🔿	Country
AttractionId AttractionName	P Countryld CountryName
AttractionDescripti CountryId CategoryId AttractionPhoto	
CityId	CountryCity 🛞
AttractionAddress	Countryld Cityld CityName
Category 🛞	
CategoryId CategoryName	

der Countryld, Cityld

Name, AttractionPhoto, etc.

We need, as in the case of a listing, that both For each commands (either implicit, i.e. coming from grids with a base table, or explicit, i.e. coming from grids without a base table) have the same base table, Attraction. And that the first one makes up the group that is going to make the break by country/city.

To do so, it is enough to modify the base transaction and add the order to the first grid.

It will depend on whether grids were implemented with or without a base table, to see how to do it.

WITH Base Tables

Let's start by the case in which the web panel was implemented with both grids with a base table.

With several Grids: nested Attraction -Country (2) AttractionId AttractionName P Countryld CountryName AttractionDescripti CountryId CategoryId AttractionPhoto CountryCity CityId AttractionAddress Countryld Cityld CityName Properties -Web Form Rules Events Conditions Variables × General Class 1 □ parm(in: CountryId); Category 1 📜 🛃 🌾 Filter × ^ Categoryld CategoryName Free Style Grid: Grid1 Control Name Grid1 Collection Country Name CountryName Rendering Mode Responsive For each Attraction order Countryld, Cityld Save State False GRID Base Trn Attraction print CountryName City Name CityName Order Countryld, Cityld print CityName Conditions GRID For each Attraction Unique Attraction Id Attraction Name print AttractionName, AttractionPhoto, etc. irips AttractionId AttractionName &trips &update2 &newTrip endfor C endfor

In this case, the objective is achieved by changing the properties that we see of the external grid, for these others.

Jenno	Properties		P			
ViewCountryInfo_relatedCopy1 ×	General Class					
Web Layout Rules Events Conditions Variables Help Documentation	📜 🛃 🌾 Filter					
I I I I I I I I I	✓ Free Style Grid: Gri	d1				
<no action="" group="" selected=""></no>	Control Name	Grid1				
MainTable Grid1	Collection					
Country Name CountryName	Rendering Mode	Responsive				
	Save State	False			-	
GRID	Base Trn	Country.City	Properties General Class		7	^
City Name CityName	Order		Filter			×
GRID	Conditions		→ Grid: Grid2			
Attraction Id Attraction Name Trips	Unique		Control Name	Grid2		
AttractionId AttractionName & ktrips & update2 & newTri	ip		Collection			
			Base Trn	Attraction		
			Order	AttractionName		
Total Trips &totalTrips			Conditions			
٥ <u>ــــــــــــــــــــــــــــــــــــ</u>			Unique			
Total Attractions &totalAttractions			Data Selector	(none)		
				(1010)		

Before doing it, let's go to GeneXus. We see the properties of both grids: the first one has a Base Trn Country.City, and no order.

The second one has Base Trn Attraction and an order by AttractionName.

ViewCountryInfo_relatedCopy1 ×	
Web Layout Rules Events Conditions Variables Help Documentation	
No action group selected>	
✓ Ⅲ MainTable ■ Grid1	
Country Name CountryName	
GRID	9
City Name CityName	
GRID	
Attraction Id Attraction Name Trips	
AttractionId AttractionName & divergence & divergence <td< td=""><td></td></td<>	
Total Trips &totalTrips	
Total Attractions & & total Attractions	

Event Grid1.Refresh &totalAttractions = 0 endevent
Event Grid2.Refresh &totalTrips = 0 Endevent
<pre>Event Grid1.Load &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions Endevent</pre>
<pre>Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent</pre>
Event <pre>&update2.Click Attraction(trnMode.Update, AttractionId) Endevent</pre>
<pre>Event &newTrip.Click &trips = NewTrip(AttractionId) Refresh endevent</pre>
Event AttractionName.Click ViewAttractionFromScratch(AttractionId) Endevent

And in the events we see that the Refresh and Load events have been programmed for each grid, only to initialize and sum or count in variables (which will show the total number of attractions loaded and the total number of trips in which those attractions are included).

We also have user events to call various objects. They don't matter at all here.



In GeneXus, let's look at the navigation list of the web panel.

We can clearly see the base table of the first grid: CountryCity, which will filter by the country received in a parameter. Next, we go to the nested Load, which has Attraction as a base table, and filters by the country and city of the external grid.

Of course, we see the join

Also, note that it orders the first implicit For each by Countryld, and the second one by AttractionName, for which it reports that there is no index.

	<pre>Event Grid1.Refresh</pre>
Web Form Rules Events Conditions Variables	For each Country.Clty where CountryId = @CountryId Event Grid1.Load &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions
Country Name CountryName GRID GRID GRID GRID	endevent Load Event Grid2.Refresh &totalTrips = 0 Endevent
Attraction Id Attraction Name Trips AttractionId AttractionName & wewTrip AttractionId AttractionName & wewTrip Total Trips & totalTrips	For each Attraction order AttractionName where Countryld = @Countryld where Cityld = @Cityld Event Grid2.Load
Total Attractions &totalAttractions Grid1 and Grid2 with Base Tables	<pre>&trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent Load endfor endfor</pre>

The pseudocode of the source that GeneXus will program will be similar to the one displayed. In it, the Base Transaction property of Grid1 was used to program the base Transaction of the implicit For each.

In the second For each, the order clause set was the content of the grid's Order property, which was AttractionName, and that is why we saw those selected indexes.

In short, the Refresh of the external grid will be triggered and then the implicit For each that we are seeing, which together with the internal one will make up a **join**.

				<pre>&totalAttractions = 0 endevent</pre>
(S Properties Seneral Class ⊉ ∑ Filter		₽ × ×	For each Attraction order Countryld, Cityld where Countryld = @Countryld Event Grid1.Load &attractions = Count(AttractionName)
Country Name CountryName	Free Style Grid: Grid:	d1	^	&totalAttractions = &totalAttractions + &attractions endevent
GRID GRID GRID GRID	Control Name Collection Rendering Mode Save State	Grid1 Responsive False		Load Event Grid2.Refresh &totalTrips = 0 Endevent
Attraction Id Attraction Name AttractionName	Base Trn Order Conditions	Attraction Countryld, Cityld		For each Attraction order AttractionName where CountryId = @CountryId where CityId = @CityId
Total Trips &totalTrips Total Attractions &totalAttractions Grid1 and Grid2 with Base Table	Unique			Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent Load endfor endfor

Contration Contration Contracts

Now, let's modify the properties of Grid1, the external one.

In doing so, the Base Transaction property of the grid will cause the Base Transaction of the implicit For each to be modified. And the Order property will become the order clause of that For each.

This will be enough, because the Refresh of the external grid will be triggered and then the implicit For each that we are seeing, which together with the internal one will make up a **control break**. Therefore, tourist attractions are going to be grouped by cities of the country (the filter by Countryld is due to the parameter).

Therefore, for **each group** of tourist attractions formed by each city, the Load event will be triggered once. Then the first line will be loaded in the grid, with the CityName of the city of the first group of attractions (and of course, its CountryName, which will be the same for all of them).

Next, the Refresh event of the nested grid will be triggered, after which the implicit for each will be executed, and it will only run through the attractions of the group, i.e., those of <u>that</u> country and city. For each one of these attractions in the group it will trigger the Load event and the Load command to load it into the nested grid.

And so on with all the groups.

GeneXus

Demo	Web Panel ViewCountryInfo_relatedCopy1Copy1 Navigatio	n Report *					
Demo	Name: ViewCountryInfo_relatedCopy1Copy1 Environment: Description: View Country Info_related Copy1 Copy1 Spec. Version: Form Class: Program Name Parameters: Parameters:						
-	Warnings A spc0038 There is no index for order <u>Countryld, Cityld, AttractionName</u> ; po 'Grid2'.	bor performance may be noticed in grid					
-	Event Grid1.Load	Event Grid2.Load					
	Order: Countryld, Cityld, AttractionName No indext Navigation filters: Start from: Countryld = @Countryld Join location: Server Image: Countryld / INTO Cutyld AttractionPhoto AttractionPhotos Image: Countryld / INTO Cutyld AttractionPhoto AttractionPhotos Image: Countryld / INTO Cutyld / INTO Citylame Image: Countryld / INTO Cutyld / INTO Citylame Image: CountryCity (Countryld / Cityld / INTO Citylame Image: Countryld / Into Cutyld / Into Citylame Image: Countryld / Into Cityla / INTO Cutyld / Into Citylame Image: Countryld / Into Cutyld / Into Cityla Index: INTRACTION Group by: AttractionId Countryld Cityld Index: INTRACTION Group by: AttractionId Countryld Cityld	Event Grid 1 Order: No indext Navigation filters: Loop While: Join location: Server Image: Server CountryId = @CountryId and CityId = @CityId Image: Server Image: Server Image: Server Server Image: Server Count(TripDate) navigation (AttractionPhoto AttractionPhoto Uri, AttractionName AttractionId Image: Server Server Server Server <t< td=""></t<>					

If we look at the navigation list, we can clearly see that both grids will navigate the same table, Attraction, using an index made up of the attributes of the Order property of the first grid, plus the order of the nested grid.

And we can clearly see that for the nested grid only the attractions corresponding to the country and city of the first grid will be run through.

					Recents	Countries — View C	ountry Info				
Trave	l Agency				COUNTRY NAME	France					
Recents	Countries — View Co	puntry info				City Name	Paris				
COUNTRY NAME	China					Attraction Name			Trips		
	City Name	Beijing				Eiffel Tower		A	5	UPDATE	NEW TRIP
	Attraction Name		Trips			Louvre Museum		as at	1	UPDATE	NEW TRIP
	Forbidden city		0	UPDATE							
	Meet the Emperor		0	UPDATE		Total Trips City Name	6 Nice				
	The Great Wall		0	UPDATE		Attraction Name			Trips		
	Total Trips	0				Matisse Museum			2	UPDATE	NEW TRIP
Total Attractions		1									
					Total Attractions	Total Trips	2				

If now we try it at runtime... we see exactly what we wanted.

Take China, for example. Perfect.

And if we go to France... in this case we don't notice any differences with the case of a join.

But the attractions are not being counted correctly. Why?

	Event Grid1.Refresh &totalAttractions = 0 endevent
Web Form Rules Events Conditions Variables 1 parm(in: CountryId);	For each Attraction order Countryld, Cityld where Countryld = @Countryld Event_Grid1.Load
Country Name CountryName GRID City Name CityName	& attractions = Count(AttractionName) & attractions = & totalAttractions + & attractions endevent Load Event Grid2.Refresh & totalTrips = 0
GRID Attraction Id Attraction Name AttractionName Constraints AttractionNam	Endevent For each Attraction order AttractionName where Countryld = @Countryld where Cityld = @Cityld Event Grid2.Load
Total Attractions &totalAttractions Grid1 and Grid2 with Base Tables	<pre>&trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent Load endfor endfor</pre>

In Grid1 we are using the Count formula to count the attractions corresponding to that country and city. This worked when the base table of the For each was CountryCity, but not now that it is Attraction.

Demo	Web Panel ViewCountryInfo_relatedCopy1Copy1 Navigat	on Report *
	Name: ViewCountryInfo_relatedCopy1Copy1 Environmen Description: View Country Info_related Copy1 Copy1 Spec. Versic Form Class: Program Na Parameters:	n: <u>17_0_3-148529</u> HTML Grid1 and Grid2 with Base Tables
	Warnings	*
	spc0038 There is no index for order <u>Countryld</u> , <u>Cityld</u> , <u>AttractionName</u> ; 'Grid2'.	poor performance may be noticed in grid
	Event Grid1.Load	Event Grid2.Load *
	Order: Countryld, Cityld, AttractionName Navigation filters: Start from: Countryld = @Countryld Join location: Server Countryld = @Countryld Image: Attraction (Attractiond) INTO Cityld AttractionPhoto AttractionPhoto Image: AttractionPhoto AttractionPhoto Image: Country Countryld (INTO Countryld) INTO CitylAame Image: Countryld (Intro CitylAame	Grid1 Order: <u>Countryld</u> , Cityld, AttractionName No index! Navigation filters: Loop while: <u>Countryld</u> = @Countryld and Cityld = @Cityld Join location: Server d #=Attraction (AttractionId) INTO AttractionPhoto AttractionPhoto.Uri. AttractionName AttractionId #=count(TripDate.) navigation (AttractionId)
	=count(AttractionName).navigation (AttractionId, Country	Formulas *
	Formulas Navigation to evaluate: count(<u>AttractionName</u>) Given: <u>AttractionId CountryId CityId</u> Index: <u>AttractionId CountryId CityId</u> Group by: <u>AttractionId CountryId CityId</u>	Navigation to evaluate: count(<u>TripDate</u>) Given: <u>AttractionId</u> Index: IATTRACTION Group by: <u>AttractionId</u>

The problem with the formula of the first grid is clear in the navigation list.

We can't run through a table and make an aggregation on the same table.

GeneXus

				Properties		- 7 - ×		
Web Panel ViewCountryInfo_relatedCopy1Co	Web Panel ViewCountryInfo_relatedCopy1Copy1 Navigation Report *							
				📜 🛃 🌾 Filter				
Name: ViewCountryInfo_relatedCopy1Copy1 Description: View Country Info_related Copy1 Copy1	Environment: Spec. Version			✓ Free Style Grid: Grid	d1	^		
	Form Class:	HTML		Control Name	Grid1			
	Program Nam	/ = // //		Collection				
	Parameters:	in: <u>Countryld</u>		Rendering Mode	Responsive			
Varnings			*	Save State	False			
spc0038 There is no index for order Countryld, Cityld, J	AttentionNemo			Base Trn	Attraction			
'Grid2'.	AttractionName, p	boor performance may be noticed in grid		Order	Countryld, Cityld	-		
Event Grid1.Load		Event Grid2.Load		Conditions				
Order: Countryld , Cityld , AttractionName No index! Navigation filters: Start from: Countryld = @Countryld Loop while: Countryld = @Countryld Join location: Server =-Attraction (<i>AttractionId</i>) INTO Cityld AttractionPh =-Country (<i>Countryld</i>) INTO CountryName	2	Grid1 Countryld , Cityld , AttractionName No indext No indext Join location: Server Image: Attraction (AttractionPhoto AttractionPhoto AttractionPhoto.Uri AttractionName AttractionId) Image: Attraction (Attraction (AttractionAttractionPhoto AttractionPhoto.Uri AttractionName AttractionId) Image: Attraction (AttractionAttractionPhoto AttractionPhoto AttractionName AttractionId)						
		G Formulas Navigation to evaluate: count(<u>TripDate</u>)						
Formulas								
Navigation to evaluate: count(<u>AttractionName</u>) Given: <u>Attractionid Countryld Cityld</u> Index: <u>IATTRACTION</u>		Given: <u>AttractionId</u> Index: IATTRACTION Group by: <u>AttractionId</u>						
Group by: <u>AttractionId CountryId CityId</u>								
E=Attraction (AttractionId, CountryId, CityId)	error spc021 Failed: Spec	 11: Unique clause in break group n cification	ot suppo	orted in grid 'Gridl'.	. (Web Panel 'ViewCountryIn	fo_rel:		

For this to work we would have to use the **unique clause**, which in this case is not useful, because it is not supported for control breaks (we're talking about the control break between grid 1 and grid 2).

City Name Ci	tyName				
GRID					
Attraction Id AttractionId	Attraction Name AttractionName		Trips &trips	&update2	&newTrip
Total Trips &	otalTrips				
otal Attractions	&totalAttractions]			

Event Grid1.Refresh &totalAttractions = 0 endevent
For each Attraction order Countryld, Cityld
where CountryId = @CountryId
Event Grid1.Load &attractions = 0 for each Attraction &attractions = &attractions + 1
endfor
<pre>&totalAttractions = &totalAttractions + &attractions endevent</pre>
Load
Event Grid2.Refresh &totalTrips = 0 Endevent
For each Attraction order AttractionName
where CountryId = @CountryId
where CityId = @CityId
Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent
Load
endfor
endfor

Therefore, we could make this calculation with another For each, by implementing another control break nested to the outermost For each.

But this is not what we need. We would have **one** control break **split** in two instances, but the first one would run through all the attractions of the city, and the second one would have no attractions to run through.

	Event Grid1.Refresh &totalAttractions = 0 endevent
Country Name CountryName	For each Attraction order Countryld, Cityld where Countryld = @Countryld
GRID GRID Attraction Id Attraction Name AttractionId AttractionName Total Trips &totalTrips Total Attractions &totalAttractions	Load Event Grid2.Refresh &totalTrips = 0 Endevent For each Attraction order AttractionName where Countryld = @Countryld where Cityld = @Cityld Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalAttractions + 1
Grid1 and Grid2 with Base Tables	Endevent Load endfor endfor

We have a much simpler solution: since the number of attractions will be the sum of all the records loaded in Grid 2, we could have deleted the Load event from Grid 1, and added the sum to the Load of Grid 2, without reinitializing the variable other than in the Refresh...

This makes it much simpler.

Demo							Trave	el Agency							
							Recents	Countries — View Co	untry Info						
Recents	Countries — View C	Country Info					COUNTRY NAME	Brazil							
COUNTRY NAME	France							City Name	Rio de Janeiro						
	City Name	Paris						Attraction Name			Trips				
	Attraction Name			Trips				Christ the Redemmer			0	UPDATE NEW TRI			
	Eiffel Tower		A	5	UPDATE	NEW TRIP	Total Attractions	Total Trips	0						
	Louvre Museum			1	UPDATE	NEW TRIP			Trave	I Agency				-21	
	Total Trips	6							Recents	Countries - View	Country Info_				
	City Name	Nice							COUNTRY NAME	China					
	Attraction Name		<i></i>	Trips						City Name	Beijing				
	Matisse Museum			2	UPDATE	NEW TRIP				Attraction Name			Trips		
										Forbidden city		1	0	UPDATE	NEW TR
	Total Trips	2								Meet the Emperor		5	0	UPDATE	NEW TR
Total Attractions							×			The Great Wall		_	0	UPDATE	NEW TR
										Total Trips		0			
									Total Attractions			· · · · · · · · · · · · · · · · · · ·			

If we run... for example, for France, with its two cities, no difference can be seen with the implementation with a join, because both cities have attractions.

But if we choose Brazil, we do see the difference. Or China, for example.

And now the attractions are being counted correctly.

WITHOUT Base Tables

Now let's move on to the implementation case without a base table.

/eb Form Rules Events Conditions Variables	
Country Name &CountryName	
RID City Name &cityName	
	rips ktrips &update2 &newTrip
Total Trips &totalTrips	
Total Attractions & total Attractions	

<pre>Event Grid1.Refresh &totalAttractions = 0 endevent</pre>
Event Grid1.Load For each Country.City &CountryName = CountryName &cityName = CityName Load endfor endevent
Event Grid2.Refresh &totalTrips = 0 Endevent
Event Grid2.Load For each Attraction order AttractionName where CityName = &cityName &AttractionId = AttractionId &AttractionName = AttractionName &AttractionPhoto = AttractionPhoto &trips = Count(TripDate) &totalTrips = &totalTrips + &trips &totalAttractions = &totalAttractions + 1 Load endfor
Endevent

We had this web panel that implemented the same join as in the beginning, but without base tables. Note that in the screen we only have variables and there are no attributes in the properties of any of the grids.

In the events, we explicitly performed the database loading. Let's do a Save As to leave this one as it was, with a join. And implement the control break in another. We take this opportunity to remove the Count of attractions from the first Load, and count in the second one, to make everything simpler.

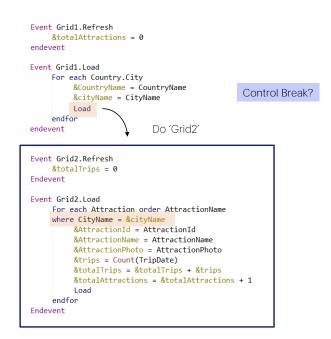
Now let's modify the Work With Country action to invoke this web panel.

Grid1 and Grid2 without Base Tables

Recents	Countries — View C	Country Info				
JNTRY NAME	France					
	City Name	Paris				
	Attraction Name			Trips		
	Eiffel Tower		4	5	UPDATE	NEW TRIP
	Louvre Museum			1	UPDATE	NEW TRIP
	Total Trips	6				
	City Name	Nice				
				D		
	Attraction Name			Trips		
	Matisse Museum		1	2	UPDATE	NEW TRIP
	Total Trips	2				
otal Attractions		3				

Let's run... and see the attractions of France. And now those of China. The **join** is clearly noticeable and not the **control break**.

Web Form Rules Events Conditions Variables					
Country Name &CountryName					
GRID					
City Name &cityName					
GRID					
Attraction Id Attraction Name &AttractionId &AttractionName &AttractionId &AttractionName Trips &update2					
Total Trips &totalTrips					
Total Attractions &totalAttractions					
Grid1 and Grid2 without Base Tables					



Even though the Load command of the first grid triggers the Refresh event, and immediately after that the Load of the second grid, it doesn't really nest the For each commands. It's as if a subroutine were invoked, as if the For each of the nested grid were executed in isolation.

That's why GeneXus is not finding an automatic join, and we had to <u>explicitly</u> filter the attractions of the city that was loaded in the &cityName variable, which was loaded by the Load event that invoked the Load of the nested grid. We didn't have to also place a filter by Countryld because it is instantiated in the parameter.

Let's keep this in mind, because it will make this case less obvious than it might seem at first.

The question is: How do we make the For each corresponding to grid1 change its base table to Attraction and establish a control break by country, city?

			Event Grid1.Refresh
Rules Events Conditions Variables	Properties	ą	
1⊡parm(in: CountryId);	General Class		endevent
	🔠 🛃 🌾 Filter)	×
0	Free Style Grid: Grid1		A Event Grid1.Load
Country Name ReCountry Name	Control Name	Grid1	For each Attraction order CountryId, CityId
Country Name &CountryName	Collection		&CountryName = CountryName
	Rendering Mode	Responsive	<pre>&cityName = CityName</pre>
GRID	Save State	False	Load
City Name &cityName	Base Trn		endfor
	Order		endevent
GRID	Conditions		
			Event Grid2.Refresh
Attraction Id Attraction Name Trips	Unique		&totalTrips = 0
&AttractionId &AttractionName O &trip	s &update2 ≠	ewTrip	Endevent
			Event Grid2.Load
			For each Attraction order AttractionName
Total Trips &totalTrips			where CityName = &cityName
			&AttractionId = AttractionId
			&AttractionName = AttractionName
Total Attractions &totalAttractions			<pre>&AttractionPhoto = AttractionPhoto</pre>
			<pre>&trips = Count(TripDate)</pre>
			<pre>&totalTrips = &totalTrips + &trips</pre>
			<pre>&totalAttractions = &totalAttractions + 1</pre>
Grid1 and Grid2 without Base Tables	2		Load
			endfor
			Endevent

It won't be by adding Base Trn or Order to the properties of Grid1 (because if we did that, we would transform the implementation into one with a base table), but to the explicit For each of the Load event of Grid1.

Therefore, it seems obvious that the first thing to do is to modify the first For each so that the Base Transaction is Attraction...

And it would also seem obvious that we should place an order clause to establish the grouping criteria by which we want the control break to be established in relation to the For each of Grid2.

Name: ViewCountryInfo_relatedCopy2Copy1 Description: View Country Info_related Copy2 Copy1	Environment: Spec. Version: Form Class:	 C# Default (C#) ▶ 17_0_3-148529 HTML
	Program Name: Parameters:	ViewCountryInfo_relatedCopy2Copy1
Varnings xpc0038 There is no index for order AttractionNa starting at line 26.	<u>me;</u> poor perform	* nance may be noticed in group
vent Grid1.Load		\$
For Each Attraction (Line: 14)		\$
Order: <u>Countryld</u> , <u>Cityld</u> Index: IATTRACTION1 Navigation Start from: <u>Countryld</u> = @ <u>Countryl</u> , filters: Loop while: <u>Countryld</u> = @Countryl		

	This is not a Control Break!	
Event Grid2.Load		*
For Each Attrac	ion (Line: 26)	*
Order: Navigation filters: Constraints: Join location: IIII-At	AttractionName No index! Start from: FirstRecord Loop while: NotEndOTable CountryId @CountryId CityName = &cityName Server raction (<i>AttractionId</i>) INTO CityId CountryId AttractionPhoto.Uti	
Attract	ionPhoto AttractionName AttractionId <u>-CountryCity</u> (<u>CountryId</u> , <u>CityId</u>) INTO <u>CityName</u> <u>-count(TripDate.).navigation</u> (<u>AttractionId</u>)	
Formulas		*

However, if we look at the navigation list...

It seems a bit odd, and although every For each apparently does what it should do, it didn't choose the same order for each one, in order to use a single index and make the run through more efficient. Something is not right.

It obviously didn't understand that it will have to make a control break.

S View Country Info	_related Copy1 × +					0	> -	- 0	×
$\leftarrow \ \ \rightarrow \ \ C \ \ \Delta$	trialapps3.genexus.com	n/ld111ff2ece4700f8312f2a	98d30e5285c/viewcountryinfe	p_relatedcopy1.aspx?CountryId=2		☆	<u>^</u> *	- 🗊 🔇	:
Recents	Countries — View Cou	ntry Info							
COUNTRY NAME	France								h
	City Name	Paris							
	Attraction Name				Trips				
	Elffel Tower		A		5	UPDATE	. N	IEW TRIP	
	Louvre Museum		as the		1	UPDATE	N	IEW TRIP	
	Total Trips		6						
	City Name	Nice							
	Attraction Name				Trips				
	Matisse Museum				2	UPDATE	: N	IEW TRIP	
	Total Trips		2						
Total Attractions		3 🔓							

And we can confirm this by running it.

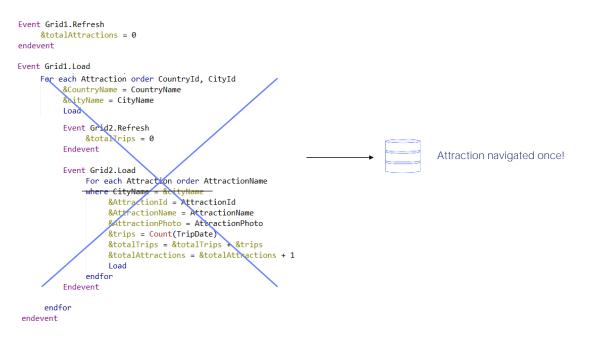
Note that for the attractions in France, Paris comes up twice, corresponding to the two Paris attractions that are available.

Trave	Agency				1							
Recents	Countries — View Cou	ntry info										
DUNTRY NAME	China											
	City Name	Beijing										
	Attraction Name			Trips								
	Forbidden city	D-	a The	0	UPDATE	NEW TRIP						
	Meet the Emperor			0	UPDATE	NEW TRIP						
	The Great Wall			0	UPDAT	Trava	Agency					
			200			Have	Agency					
	Total Trips	0								,		
	City Name	Beijing				Recents	Countries - View Co	untry Info				
						COUNTRY NAME	Brazil					
	Attraction Name			Trips								
	Forbidden city		1000	0	UPDAT		City Name	Rio de Janeiro				
	Meet the Emperor		di la constante	0	UPDAT							
	The Great Wall			0	UPDAT		Attraction Name		 Trips			
							Christ the Redemmer		0		UPDATE	NEW TRIP
	Total Trips	0					Total Trips	a				
	City Name	Beijing					Toma Tripo					
						Total Attractions		1				
	Attraction Name			Trips								
	Forbidden city		100	0	UPDATE	NEW TRIP						
	Meet the Emperor		đ	0	UPDATE	NEW TRIP						
	The Great Wall		a	0	UPDATE	NEW TRIP						
		D:										
	Total Trips	0										
tal Attractions		9										

For those in China, Beijing comes up three times, corresponding to the three Beijing attractions.

And for Brazil only once, matching the Rio attractions registered.

What's going on?



Obviously, it doesn't realize that it has to make a control break.

This is because, as we have already mentioned, it doesn't **really** nest the navigations.

```
Event Grid1.Refresh

&totalAttractions = 0

endevent

Event Grid1.Load

For each Attraction order CountryId, CityId

&CountryName = CountryName

&cityName = CityName

Load

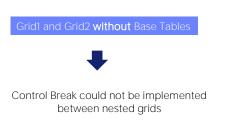
endfor

endevent
```

Ende	t Grid2.Refresh <pre>&totalTrips = 0 vent t Grid2.Load</pre>	
	For each Attraction order AttractionName	
	where CityName = &cityName	
	&AttractionId = AttractionId	
	&AttractionName = AttractionName	
	<pre>&AttractionPhoto = AttractionPhoto</pre>	
	<pre>&trips = Count(TripDate)</pre>	
	<pre>&totalTrips = &totalTrips + &trips</pre>	
	&totalAttractions = &totalAttractions + 1	
	Load	
	endfor	
Ende	vent 🚝	

It's as if they were two independent For each commands, only that from one the execution of the other is invoked, but through two separate queries to the database.

Country Name	&CountryName
GRID	
City Name &c	ityName
GRID	
Attraction Id	Attraction Name Trips
&AttractionId	&AttractionName & trips & update2 & newTrip
Total Trips &t	otalTrips
Total Attractions	&totalAttractions

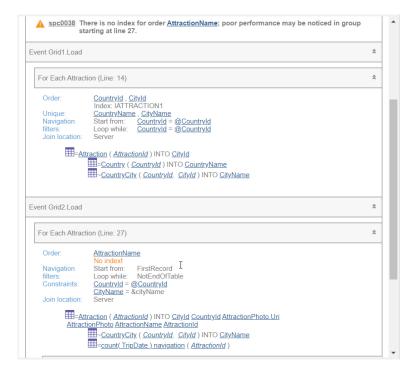


This is the equivalent to saying that we cannot **really** implement a control break between two nested grids without base tables.

RID					
City Name &	cityName				
GRID					
Attraction Id	Attraction Name		Trips		
&AttractionId	&AttractionName	<u> </u>	&trips	&update2	&newTrip
Total Trips &	totalTrips				
	&totalAttractions				

<pre>Event Grid1.Refresh &totalAttractions = 0 endevent</pre>
Event Grid1.Load
For each Attraction order CountryId, CityId unique CountryName, CityName &CountryName = CountryName &cityName = CityName Load endfor
endevent
Event Grid2.Refresh &totalTrips = 0 Endevent
Event Grid2.Load For each Attraction order AttractionName where CityName = &cityName &AttractionId = AttractionId &AttractionName = AttractionName &AttractionPhoto = AttractionPhoto &trips = Count(TripDate) &totalTrips = &totalTrips + &trips &totalAttractions = &totalAttractions + 1 Load endfor Endevent

The solution we have for the moment is to use, for the first For each, the unique clause. In other words, if several records in the Attraction table have the same country and city, it will keep only one of them. And for this one, load the variables and execute the Refresh event, and right after that the Load of grid2, which will execute its For each as if it were completely independent of the previous one. And that is precisely why this time it will allow us to use the unique clause.



If now we look at the navigation list... it seems that it will work this way.

Demo	Travel Agency	
	Recents Countries - View Country Info	
	COUNTRY NAME China	
	City Name Beijing	
Recents Countries View Country Info	Attraction Name	Trips
COUNTRY NAME France		0 UPDATE NEW TRIP
City Name Nice		0 UPDATE NEW TRIP
City Name Nice		0 UPDATE NEW TRIP
Attraction Name Trips		
Matisse Museum 2	Total Trips 0	Travel Agency
Total Trips 2	Total Attractions 3	
City Name Paris		Recents Countries - View Country Info
De .		COUNTRY NAME Brazil
Attraction Name Trips	UPDATE NEW TRIP	
		City Name Rio de Janeiro
Louvre Museum 1	UPDATE NEW TRIP	Attraction Name Trips
Total Trips 6		Christ the Redemmer 0 UPDATE NEW TRIP
Total Attractions 3		-
		Total Trips 0
		Total Attractions 1

We run it...

We have succeeded.

WITH or WITHOUT Base Tables?

Grid1 and Grid2 with Base Tables Grid1 and Grid2 without Base Tables Event Grid1.Refresh Free Style Grid: Grid1 &totalAttractions = 0 Control Name Grid1 endevent Collection Event Grid1.Load Rendering Mode Responsive For each Attraction order CountryId, CityId Save State False unique CountryName, CityName Base Trn Attraction &CountryName = CountryName Countryld, Cityld Order &cityName = CityName Load Conditions endfor Unique endevent Event Grid2.Refresh Event Grid1.Refresh &totalTrips = 0 &totalAttractions = 0 Endevent endevent Event Grid2.Load Event Grid2.Refresh For each Attraction order AttractionName &totalTrips = 0 where CityName = &cityName Endevent &AttractionId = AttractionId &AttractionName = AttractionName Event Grid2.Load &AttractionPhoto = AttractionPhoto &trips = Count(TripDate) &trips = Count(TripDate) &totalTrips = &totalTrips + &trips &totalTrips = &totalTrips + &trips &totalAttractions = &totalAttractions + 1 &totalAttractions = &totalAttractions + 1 Endevent Load endfor Endevent

So, for the moment, we are finding it much easier to implement a control break when grids **have a base table**.

Strictly speaking, it will be a true control break only in that case.

In the second case, when grids don't have a base table, we are just simulating it. Actually, there will be two independent queries to the Attraction table and not a single one that solves everything, as it happens in the real control break.

We encourage you to try everything we have seen.



training.genexus.com wiki.genexus.com training.genexus.com/certifications