



## Base Tables & Navigation

Behavior Development

*GeneXus™ 16*

## Base Tables & Navigation

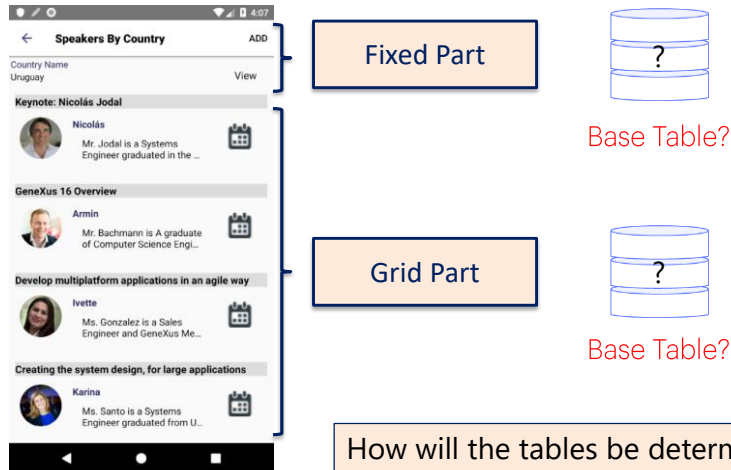


In this video we will study the specificities of base tables and navigations in objects for Smart Devices.

For a better comprehension of the topic, we will consider a simple example where we will develop a query using a Panel for Smart Devices that will enable us to view a country's data. Additionally, we will be able to view that country's speakers grouped by conference.

Then we will make a comparison against a similar web panel in order to understand the differences in the determination of base tables and navigation lists.

## Base Tables & Navigation



If we analyze this panel we will see that it will comprise a fixed part with the name of the country, the button to view it, and –in the Application Bar– the Add button to add a new speaker.

And it will also have the part corresponding to the grid.

The grid is similar to the grid of the List of the Speakers WorkWith. Here, instead of the company name we show the résumé. When we tap on the Item we access the View, as in the list of Speakers that we already know, except that, here, we program it manually. We will see that later.

And we also added an icon to view the detail of conferences.

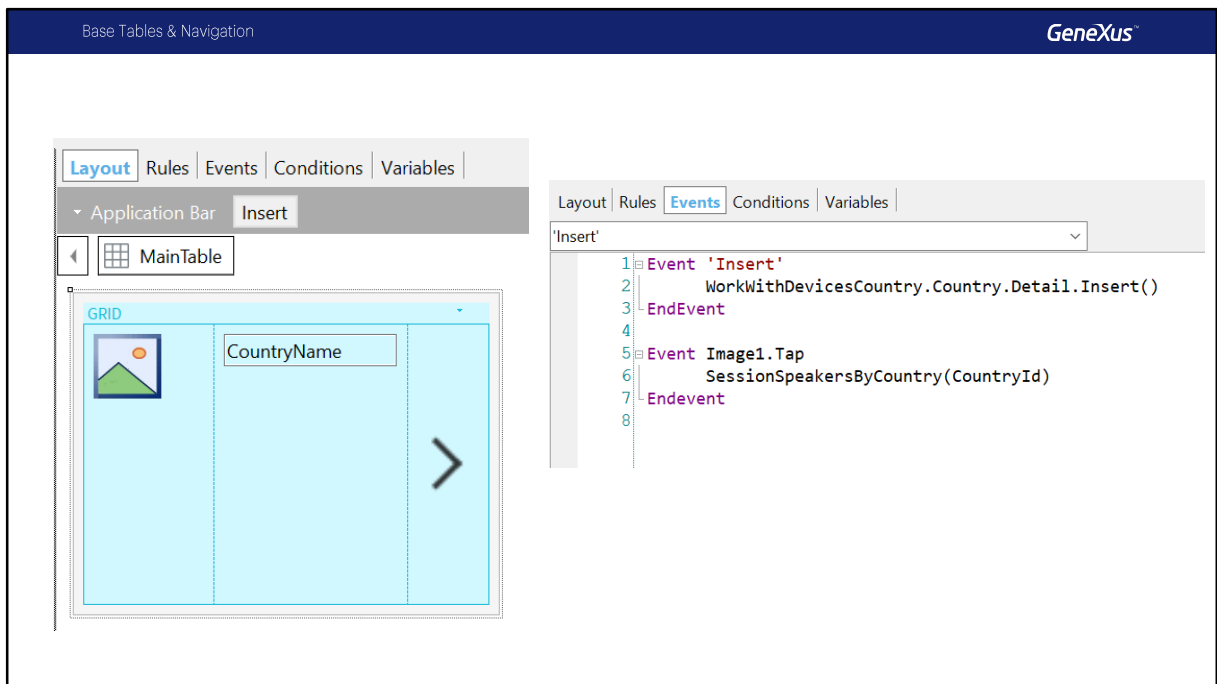
In the video, where we mentioned events, we saw that navigations are separated and are solved with data providers, one for each part of the panel.

So, which tables will be navigated in each case? And how will each of these base tables be determined?

That's what we will see next.

## Demo: Base Tables & Navigation

Let's see the example in GeneXus.



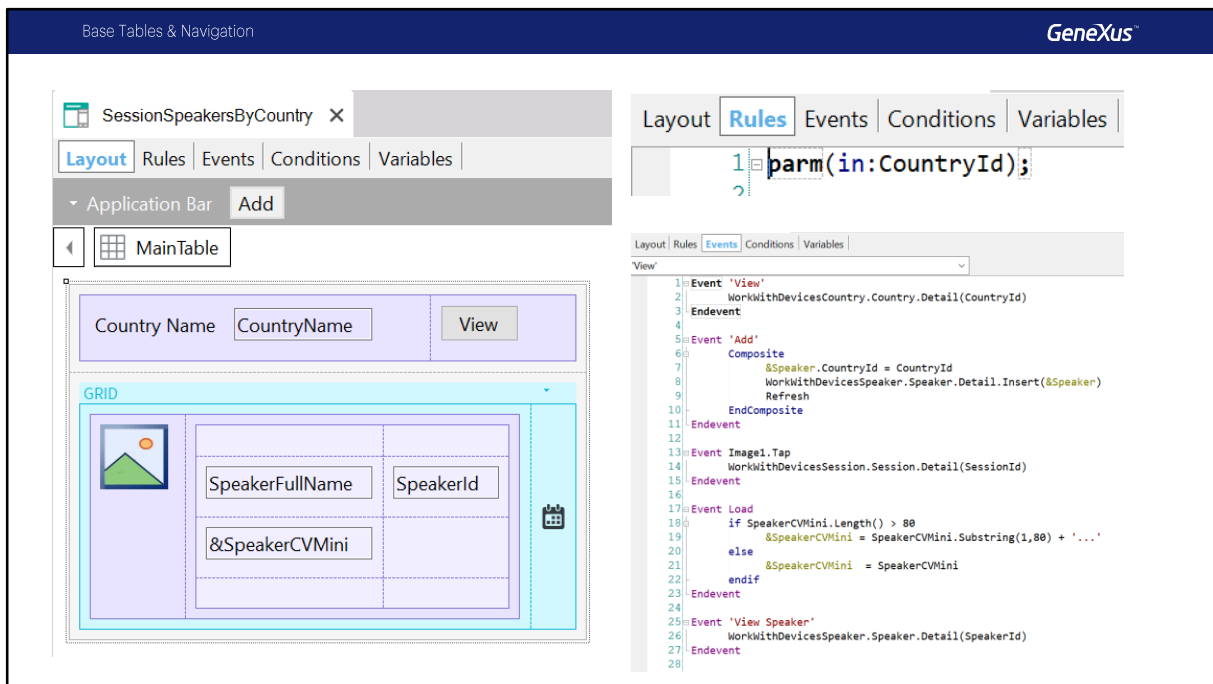
Let's quickly review the application in runtime.

We go to Country, and for example, in Uruguay, we access the panel with the new icon to view only Uruguay's speakers.

And here we can view the conferences, and the speakers of each conference. We may access a speaker and view the detail, or we may access a conference and its detail screen. We have the Add button to add a Speaker to that country, and the View button to view the country's details.

Let's first see how we called the SessionSpeakersByCountry Panel.

On the List of the WorkWithDevicesCountry we added an image on the grid, and in the TAP event we made the call to the panel, as we will see, passing the CountryId attribute by parameter.



Now, let's take a look at the panel.

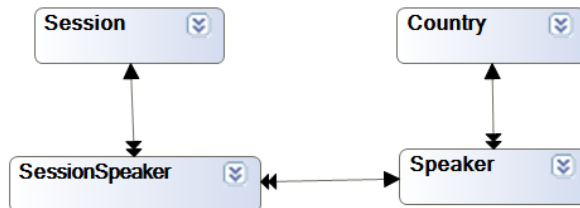
First note that in the rules, we indicated that this panel will receive the CountryId attribute by parameter.

In the layout we see that in the fixed part we have the CountryName attribute, and the View button. If we look at the associated event we see that it is calling the Country Detail and it also passes the CountryId as parameter.

On the other hand, also in the fixed part of the panel, we have the application Bar, with the Add associated button. Let's see this event.

What we do in this Event is to use a &Speaker variable that is based on the Speaker Business Component. We assign the CountryId, the one received by parameter so we initialize the Speaker country. Then we call the Detail of Speaker in Insert mode and we pass the &Speaker variable as parameter.

And we also have the grid, that will have its own base table, independently from the base table of the fixed part.



We should remember that this is a specificity of the Smart Device objects: fixed part and grid determine independent navigations. Each will have its own base table.

This defines an important difference as compared to web panels. So, in Smart Devices, we must remember that the Refresh event will navigate the base table of the fixed part. And the Load event will be in charge of navigating the base table corresponding to the grid.

If we look at the entity relation diagram in our model (here we only see some tables, the ones we will be using for this video), we will see that when the base table is Country, the extended table will include Country only, and we don't have N to 1 relations.

If we analyze the one for Session, we see that we only have Session.

If we analyze the one for SessionSpeaker, we have SessionSpeaker, Speaker, and indirectly Country and also Session.

And last, the one for Speaker will be Speaker and Country.

If you want to refresh these concepts or read more about them you may view related videos on the Wiki or those included in the GeneXus Course.





Let's get back to the panel where we will now think how to determine the base table of the fixed part of this panel.





We will consider the attributes belonging to the fixed part (in this case only CountryName), and all the attributes belonging to the events associated with the fixed part (in this case the View and Add events) outside of a For Each command.

Therefore, in our case, the CountryId attribute is used in both events, and together with the attribute of the fixed part of the layout, they are the ones that will determine the base table of this fixed part.

If we had programmed the Refresh event (it is not the case), then the attributes appearing there –which were not inside a For Each– would also be part of the determination of the base table for this fixed part.

The attributes of the Conditions Tab would also participate, though in this case there aren't any.

That's why the base table will then be Country.

Data Provider SessionSpeakersByCountry_Level_Detail Navigation Report	
<b>Name:</b>  SessionSpeakersByCountry_Level_Detail <b>Description:</b> SessionSpeakersByCountry_Level_Detail <b>Output Devices:</b> None	<b>Environment:</b>  Default (C#) <b>Spec. Version:</b>  16_0_1-129648 <b>Form Class:</b> HTML <b>Program Name:</b> SessionSpeakersByCountry_Level_Detail <b>Parameters:</b> in: <u>CountryId</u> , in: &gxid, out: SessionSpeakersByCountry_Level_DetailSdt
LEVELS	
For First Country (Line: 6)	
<b>Order:</b> <u>CountryId</u> Index: ICOUNTRY1 <b>Navigation filters:</b> Start from: <u>CountryId</u> = @CountryId Loop while: <u>CountryId</u> = @CountryId <b>Optimizations:</b> First 1 record(s)	
 =Country ( <u>CountryId</u> ) INTO CountryName	

Let's get back to the panel, and now let's think of how to determine the base table of the fixed part of this panel.

We will consider the attributes belonging to the fixed part (in this case, CountryName only), and all the attributes belonging to the events associated with the fixed part (in this case, the View and Add events) outside of a For Each command.

Therefore, in our case, the CountryId attribute will be used in both events, and together with the attribute of the fixed part of the layout, they will be the ones to determine the base table of this fixed part.

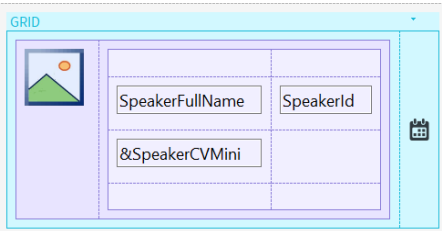
If we had programmed the Refresh event (it is not the case), then the attributes appearing there –which were not inside a For Each– would also be part of the determination of the base table for this fixed part.

The attributes of the Conditions Tab would also participate, though in this case there aren't any.

That's why the base table will then be Country.

To confirm this, let's take a look at the navigation list. If we search for the Data Provider that corresponds to the fixed part, we will see that it is actually navigating the Country table, taking the corresponding country that was received by parameter.

Base Tables & Navigation
GeneXus™



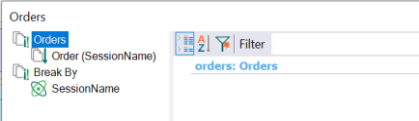
```

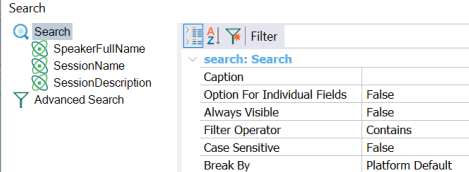
Event Image1.Tap
  WorkWithDevicesSession.Session.Detail(SessionId)
Endevent

Event Load
  if SpeakerCVMini.Length() > 80
    &SpeakerCVMini = SpeakerCVMini.Substring(1,80) + '...'
  else
    &SpeakerCVMini = SpeakerCVMini
  endif
Endevent

Event 'View Speaker'
  WorkWithDevicesSpeaker.Speaker.Detail(SpeakerId)
Endevent

```





Conditions	
Base Trn	<b>Session.Speaker</b>

We also have to identify the base table of the grid part. Of course, the attributes belonging to the grid will be part of it, in this case: SpeakerImage, SpeakerFullName, and SpeakerId. We also have the attributes of the Load event that are outside the For each: here we have the SpeakerCVMini attribute, that is used several times in the event.

And additionally, we have the event associated with the image, and the event associated with the Grid when we Tap on the element. It is the ViewSpeaker event.

Let's now see the attributes used in these events.

In the Image1.tap event we have the SessionId attribute, and in the View Speaker event we have the SpeakerId attribute.

There are more things we must take into account. For example, we have the Order of the grid. Here we have the SessionName attribute that is used to group the Speakers by Session as well.

And we have the filters in the Search property. Here we have the SpeakerFullName, SessionName and SessionDescription attributes.

And also, we have the Base Trn property, where we indicate Session.Speaker. The Conditions at the grid level or at the level of the Conditions tab –general for the whole object– are also part of this. We don't have conditions and we don't use any attribute in either of them.

In sum, we have attributes of the Speaker (Id, Image, Full Name, and Résumé) table, and we

have attributes of the Session (Id, Name, Description) table.

The base table will then be SessionSpeaker, because, from its extended table, we obtain all the attributes we need.

Data Provider SessionSpeakersByCountry_Level_Detail_Grid1 Navigation Report			
<b>Name:</b> SessionSpeakersByCountry_Level_Detail_Grid1		<b>Environment:</b> Default (C#)	
<b>Description:</b> SessionSpeakersByCountry_Level_Detail_Grid1		<b>Spec. Version:</b> 16_0_1-129648	
<b>Output Devices:</b> None		<b>Form Class:</b> HTML	
		<b>Program Name:</b> SessionSpeakersByCountry_Level_Detail_Grid1	
		<b>Parameters:</b> in: <u>CountryId</u> in: &SearchText, in: &start, in: &count, in: &grid, out: SessionSpeakersByCountry_Level_Detail_Grid1Std	
Warnings			
<div><div></div><div>sps0053 Conditional constraint "SpeakerFullName.toupper() like "%* + &amp;SearchText.toupper() or SessionName.toupper() like "%* + &amp;SearchText.toupper() or SessionDescription.toupper() like "%* + &amp;SearchText.toupper()]" cannot be generated in grid Grid1. Changed to standard constraint.</div></div>			
LEVELS			
For Each SessionSpeaker (Line: 4)			
Order: SessionId SpeakerId			
Navigation filters: Index: CHARLACRADOR FirstRecord NotEndOfTable			
Start from Loop while: CountryId = @CountryId			
Constraints: &SearchText.isEmpty() or (SpeakerFullName.toupper() like "%* + &SearchText.toupper() or SessionName.toupper() like "%* + &SearchText.toupper() or SessionDescription.toupper() like "%* + &SearchText.toupper()"			
Join location: Server			
Optimizations: Server Paging			
<div><div></div><div>--SessionSpeaker ( SessionId SpeakerId ) INTO SessionId SpeakerId</div><div></div><div>--Session ( SessionId ) INTO SessionName SessionDescription</div><div></div><div>--Speaker ( SpeakerId ) INTO CountryId SpeakerFullName SpeakerImage.Url, SpeakerCvMini SpeakerImage SpeakerSurname SpeakerName</div></div>			

If we now take a look at the navigation of the Data Provider corresponding to the grid, we will see that it is actually doing a For Each to the SessionSpeakers table, and it is filtering by country. What country? The one received by parameter.

We should keep in mind that navigations of the fixed part and of the grid are independent from one another. Filters are not established automatically between them.

What causes, in the first navigation, in the fixed part, and also in the grid, the filtering to be done by CountryId and also by Country is the fact that the CountryId attribute was received as parameter, but not the fact that the base tables are related.

Base Tables & Navigation
GeneXus™

### Web Panel WebSessionSpeakersByCountry Navigation Report

<b>Name:</b> WebSessionSpeakersByCountry <b>Description:</b> Web Session Speakers By Country	<b>Environment:</b> Default (C#) <b>Spec. Version:</b> 16_0_1-129648 <b>Form Class:</b> HTML <b>Program Name:</b> WebSessionSpeakersByCountry <b>Parameters:</b> in: CountryId
---	--

**Event Load**

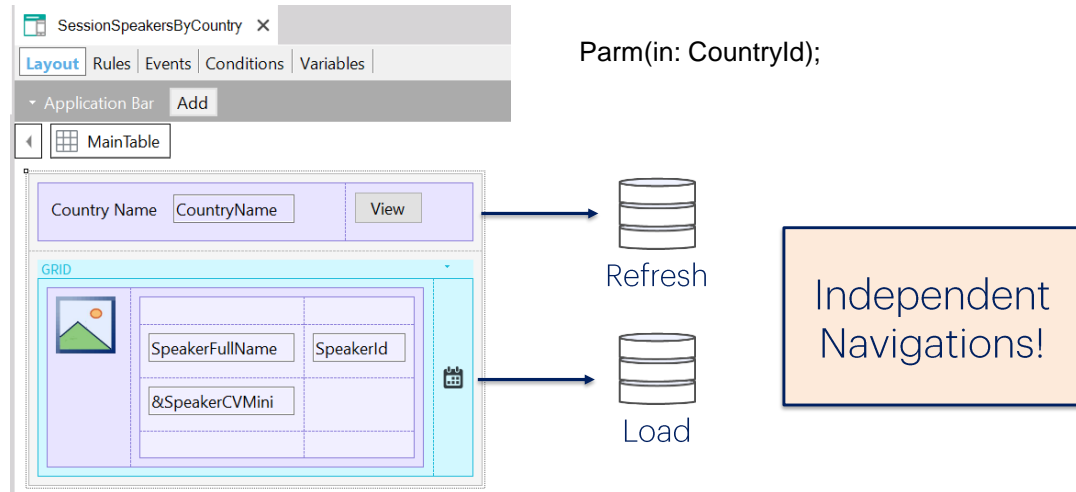
<b>Order:</b> SessionId , SpeakerId <b>Navigation filters:</b> Index: ICHARLAORADOR Start from: FirstRecord Loop while: NotEndOfTable <b>Constraints:</b> CountryId = @CountryId <b>Join location:</b> Server	<pre> --SessionSpeaker ( SessionId , SpeakerId ) INTO SessionId SpeakerId --Speaker ( SpeakerId ) INTO CountryId SpeakerCVMini SpeakerName SpeakerImage SpeakerImage.Uri --Country ( CountryId ) INTO CountryName </pre>
--	--

By way of example, and to clarify the differences between Web Panels and objects for Smart Devices, I have prepared this similar panel in a web environment. We have the CountryName attribute in the form, and the grid with the same attributes - SpeakerName, SpeakerId, the variable, SpeakerImage. This image, and the events are similar. They use the same attributes and call objects that solve the same functionality in a similar way.

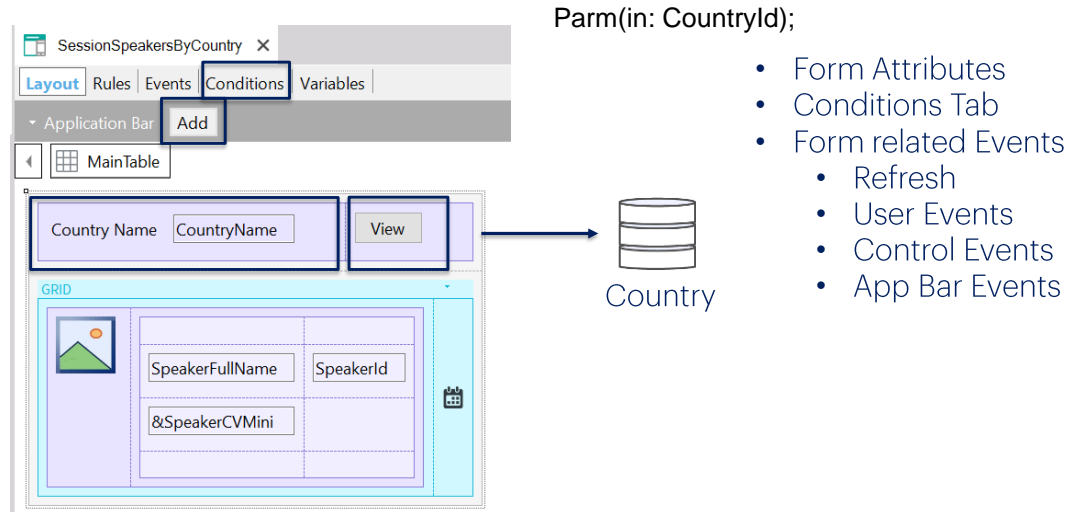
It is interesting to see the navigation. First note that the panels have a single navigation. If we take a look at the base table of the Load event, it is SessionSpeaker. From that table, the rest of the attributes necessary will be solved through the extended table.

Now let's go back to the presentation.

## Base Tables & Navigation



If we summarize what we saw on the demo: the fixed part and the grid each have their own base table, and the navigations of each are determined independently from one another.

**Base Table: Fixed Part**

The attributes that will determine the base table of the fixed part are:

- attributes included in the fixed part of the layout, in our case: CountryName
- attributes referenced in the general conditions (tab: conditions), in our case: none
- attributes appearing in the events of the form, outside For Each commands, that is: those of the Refresh, and events associated with controls of the fixed part of the form, in our case: the View button, and in the Application Bar (the Add button).

So, the base table in our case will be COUNTRY.



Base Tables & Navigation
GeneXus™

### Base Table: Grid Part

The screenshot shows the GeneXus IDE interface for an application named 'SessionSpeakersByCountry'. The 'Conditions' tab is active, and a blue arrow points from it to a table on the right. The application design shows a 'MainTable' with a 'GRID' section. A blue arrow points from the 'GRID' section to a database icon labeled 'SessionSpeaker'.

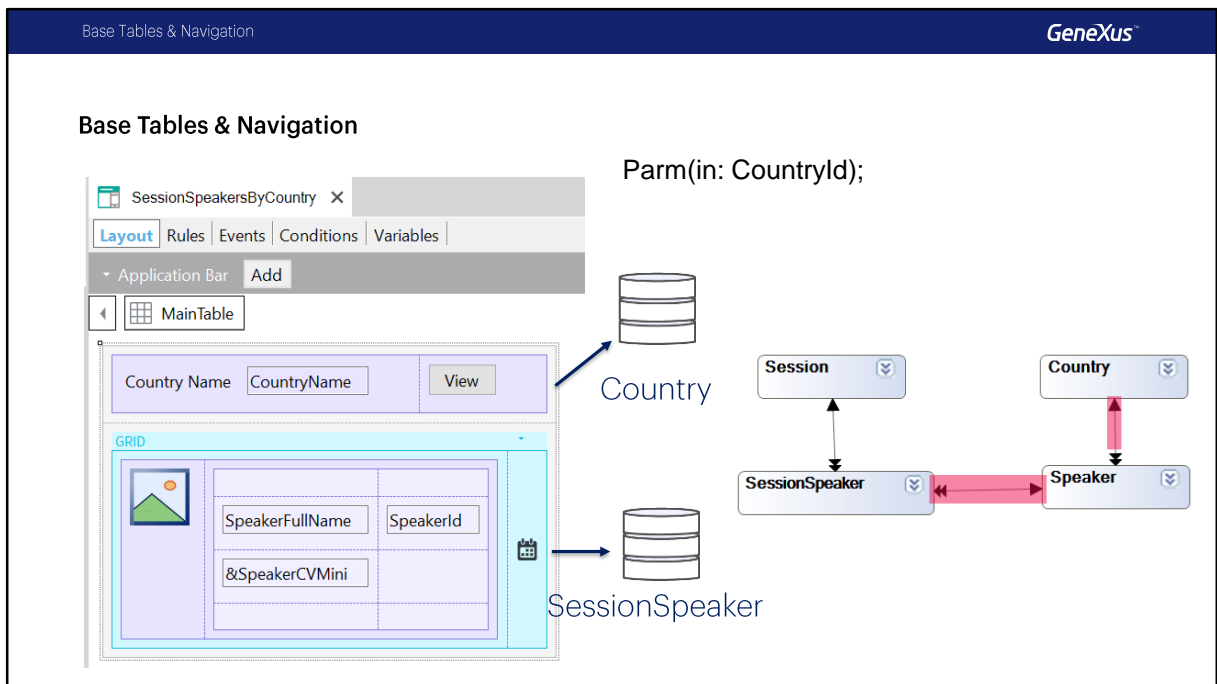
Orders	(1 order)
Search	(1 filter)
Conditions	

- Grid Attributes
- Conditions (Tab and Grid), Order, Search
- Grid related events
  - Load
  - Grid Events
  - User Grid Events

The attributes that will determine the grid's base table are:

- attributes included in the grid,
- attributes referenced in the grid's conditions, order, search and advanced search, and those of the general conditions tab.
- attributes appearing in the grid's events, outside For Each commands, that is: those of the Load and events associated with controls within the grid (in our case, that of the "ViewSession" event, and the Tap on the image).
- When the grid's Base Trn property has been set up, the base table originates here, but the previous attributes that we mentioned have to belong to the extended table for everything to be well defined.

So, in our example, the grid's base table will be SessionSpeaker.



As we already know, GeneXus identified the base tables of each part, Country and SessionSpeaker.

Despite their indirect 1 to N relation, these tables will have independent navigations by default.

This means that, it's like having a pair of parallel (non-nested) For Eachs. If we want to link the navigations we will have to do it explicitly (in this example, it was because we received in an attribute parameter).

Also note that, since the fixed part must retrieve a single record, for a well-designed panel, it must be possible to filter the corresponding base table (Country) in order to retrieve “the” record desired.

Because we are receiving by parameter in the CountryId attribute, this will occur automatically. Additionally, since the grid navigates the SessionSpeakers table—which has CountryId in its extended table—the filter will also operate on this table, and only the sessions of speakers from that country will be shown.

# GeneXus™

Videos

[training.genexus.com](https://training.genexus.com)

Documentation

[wiki.genexus.com](https://wiki.genexus.com)

Certifications

[training.genexus.com/certifications](https://training.genexus.com/certifications)

