

**GeneXus**<sup>™</sup>  
The power of doing.

# Base Tables & Navigation

Behavior Development

*GeneXus™ 16*

## Base Tables & Navigation

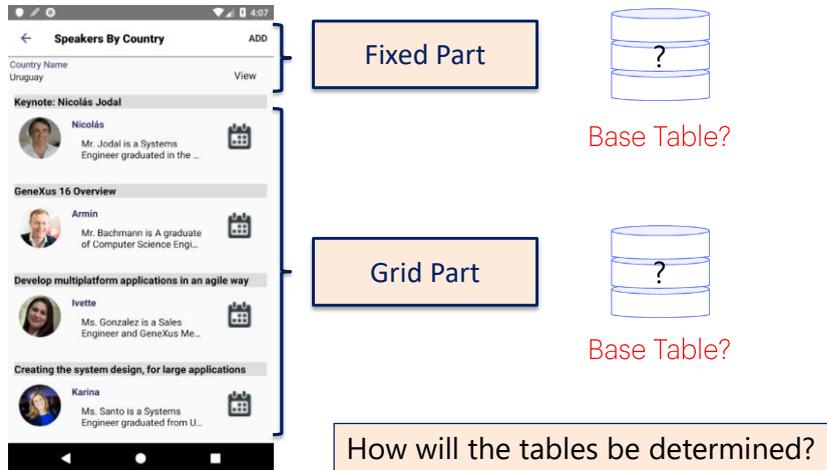


En este video estudiaremos acerca de las particularidades de las tablas base y las navegaciones en los objetos para Smart Devices.

Para entender mejor el problema, vamos a ver un ejemplo sencillo donde desarrollaremos una consulta usando un Panel for Smart Devices, este panel nos permitirá ver los datos de un país y además en un grid podremos ver los oradores de ese país agrupados por conferencia.

Luego haremos una comparación con un Panel similar en web para entender las diferencias en la determinación de las tablas base y los listados de navegación.

## Base Tables &amp; Navigation



Si nos detenemos a analizar este panel, podemos ver que va a estar compuesto de una parte fija: que contiene el nombre de país, el botón para visualizarlo, y en la Application Bar el botón Add, para agregar un nuevo orador.

Y por otro lado va a tener la parte correspondiente al grid.

Este grid es similar al grid del List del WorkWith de Speakers, aquí en lugar del nombre de la empresa estamos mostrando el Curriculum, si hacemos Tap sobre el Ítem accedemos al View, al igual que en el list de Speakers que ya conocemos, solo que acá lo programamos manualmente, ya lo veremos.

Agregamos además un icono para ver los detalles de las conferencias.

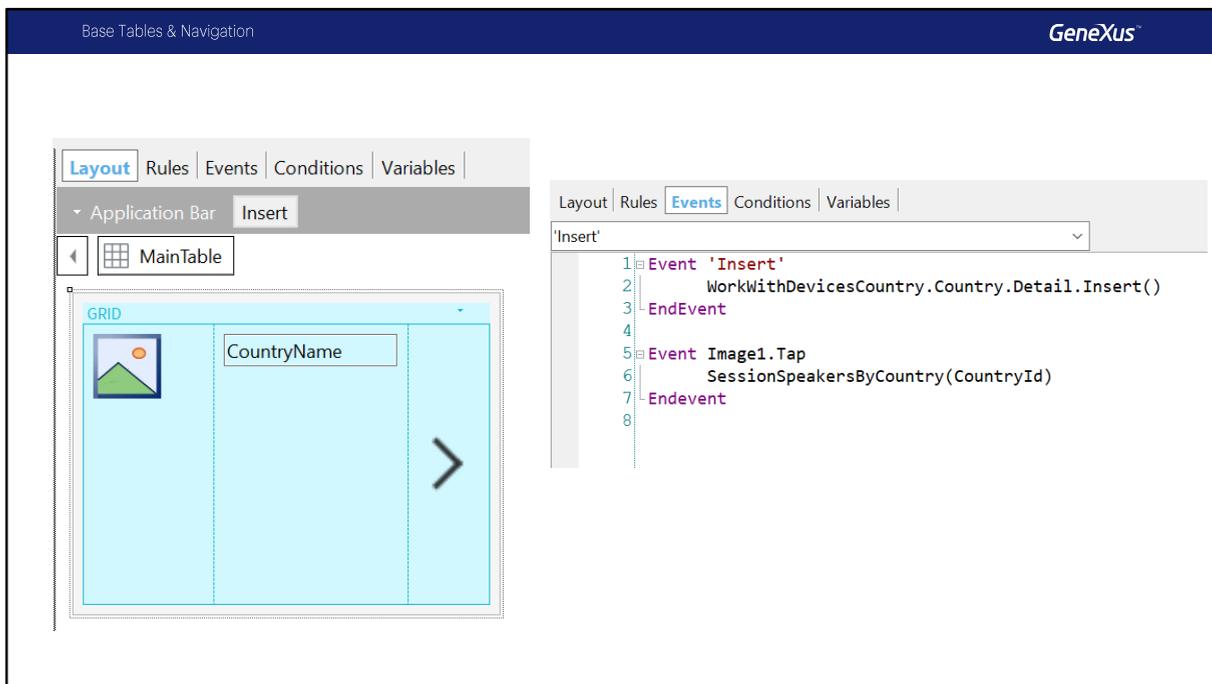
En el video donde hablamos de eventos, vimos que las navegaciones son separadas, que se resuelven con un data providers, uno por cada parte del panel.

que tablas se navegaran en cada caso? Y Como se determinara cada una de estas tablas base?

Esto es lo que estudiaremos a continuación.

## Demo: Base Tables & Navigation

Vamos a ver el ejemplo en GeneXus.



Vamos a ver el ejemplo en GeneXus.

Revisemos rápidamente en ejecución la aplicación.

Ingresamos a country, y en Uruguay por ejemplo accedemos con el nuevo icono al panel, para ver los oradores de Uruguay únicamente.

Y acá podemos ver las conferencias y para cada conferencias sus oradores, podemos acceder a un orador, estamos viendo el detalle, o podemos acceder a una conferencia, también a la pantalla de detalle.

Tenemos el botón Add para agregar un Orador a ese país y el botón View para ver los detalles del país.

Primero veamos como hicimos para llamar al Panel SessionSpeakersByCountry.

En el List del WorkWithDevicesCountry agregamos una imagen en la grilla y en el evento TAP hicimos, vamos a verlo, hicimos la llamada al panel pasando por parámetro el atributo CountryId.

Base Tables & Navigation GeneXus™

SessionSpeakersByCountry ×

Layout
Rules
Events
Conditions
Variables

Application Bar
Add

MainTable

Country Name  View

GRID

```

1 parm(in:CountryId);
2
3
4
5 Event 'Add'
6 Composite
7   &Speaker.CountryId = CountryId
8   WorkWithDevicesSpeaker.Speaker.Detail.Insert(&Speaker)
9   Refresh
10 EndComposite
11 EndEvent
12
13 Event Image1.Tap
14 WorkWithDevicesSession.Session.Detail(SessionId)
15 EndEvent
16
17 Event Load
18 if SpeakerCVMini.Length() > 80
19   &SpeakerCVMini = SpeakerCVMini.Substring(1,80) + '...'
20 else
21   &SpeakerCVMini = SpeakerCVMini
22 endif
23 EndEvent
24
25 Event 'View Speaker'
26 WorkWithDevicesSpeaker.Speaker.Detail(SpeakerId)
27 EndEvent
28

```

Veamos ahora el panel.

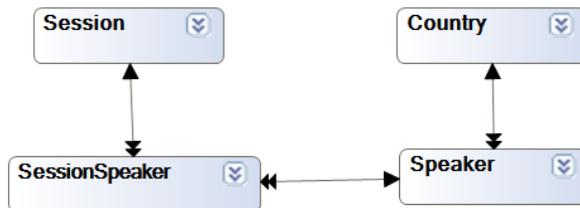
Primero notemos que en las reglas, indicamos que este panel recibirá por parámetro el atributo CountryId.

En el layout vemos que en la parte fija, tenemos el atributo CountryName, el botón View, vamos a ver el evento asociado, vemos esta llamando al Detalle del País y le pasa también el CountryId como parámetro.

Por otro lado, también en la parte fija del panel, tenemos el application Bar, que tiene el botón Add asociado, veamos este evento.

Lo que hacemos en este Evento es usar una variable &Speaker que esta basada en el Business Component Speaker, y le estamos asignando el CountryId, el recibido por parámetro así lo inicializamos al país del Orador, luego llamamos al Detalle de Speaker en modo Insert y le pasamos como parámetro la variable &Speaker.

Por otro lado, tenemos el grid, que tendrá su propia tabla base, independiente de la tabla base de la parte fija.



Recordemos que esta es una particularidad de los objetos Smart Devices: parte fija y grid determinan navegaciones independientes. Cada uno tendrá su tabla base. Esto establece una diferencia importante con los web panels. Así, en Smart devices, recordemos que el evento refresh navegará la tabla base de la parte fija. Y el evento load será quien se encargará de navegar la tabla base correspondiente al grid.

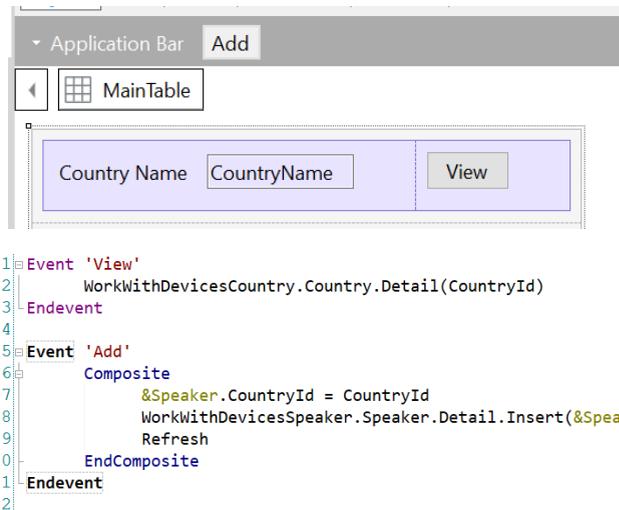
Si vemos el diagrama de entidad relación de nuestro modelo (acá solo vemos algunas tablas, las que estaremos usando en este video), podemos ver que si la tabla base es Country la tabla extendida estará compuesta por Country solamente, no tenemos relaciones N a 1.

Si analizamos la de Session tenemos Session únicamente.

Si analizamos la de SessionSpeaker, tenemos a SessionSpeaker, Speaker, indirectamente a Country y también Session.

Por último la de Speaker será Speaker y Country.

Si desean refrescar o profundizar más en estos conceptos pueden ver videos en el Curso GeneXus o en la Wiki.



Volvamos al panel y ahora si, pensemos cómo se determina, entonces, la tabla base de la parte fija de este panel.

Se tendrán en cuenta los atributos que pertenezcan a la parte fija (en este caso solamente CountryName), y todos los atributos que pertenezcan a los eventos asociados a la parte fija (en este caso los eventos View y el Add) que estén fuera de un comando for each. Por tanto, en nuestro caso, el atributo CountryId se usa en ambos eventos y junto con el atributo de la parte fija del layout, son los que van a determinar la tabla base de esta parte fija.

Si tuviéramos programado el evento Refresh (no es este el caso) los atributos que allí aparecieran –y que no estuvieran dentro de un for each- también participarían de la determinación de la tabla base de esta parte fija.

También participarían los atributos del Tab Conditions, en este caso no hay ninguno. Es por ello que la tabla base será, entonces, Country.

**Data Provider SessionSpeakersByCountry\_Level\_Detail Navigation Report** ⌵

<p><b>Name:</b> <a href="#">SessionSpeakersByCountry_Level_Detail</a></p> <p><b>Description:</b> SessionSpeakersByCountry_Level_Detail</p> <p><b>Output Devices:</b> None</p>	<p><b>Environment:</b>  Default (C#)</p> <p><b>Spec. Version:</b>  16_0_1-129648</p> <p><b>Form Class:</b> HTML</p> <p><b>Program Name:</b> SessionSpeakersByCountry_Level_Detail</p> <p><b>Parameters:</b> in: <a href="#">CountryId</a>, in: &amp;gxid, out: SessionSpeakersByCountry_Level_DetailSdt</p>
---	---

**LEVELS** ⌵

For First Country (Line: 6) ⌵

<p><b>Order:</b> <a href="#">CountryId</a></p> <p><b>Navigation filters:</b> Start from: <a href="#">CountryId = @CountryId</a> Loop while: <a href="#">CountryId = @CountryId</a></p> <p><b>Optimizations:</b> First 1 record(s)</p>	<p>Index: ICOUNTRY1</p> <p><a href="#">CountryId</a> INTO <a href="#">CountryName</a></p>
---	---

Volvamos al panel y ahora si, pensemos cómo se determina, entonces, la tabla base de la parte fija de este panel.

Se tendrán en cuenta los atributos que pertenezcan a la parte fija (en este caso solamente CountryName), y todos los atributos que pertenezcan a los eventos asociados a la parte fija (en este caso los eventos View y el Add) que estén fuera de un comando for each. Por tanto, en nuestro caso, el atributo CountryId se usa en ambos eventos y junto con el atributo de la parte fija del layout, son los que van a determinar la tabla base de esta parte fija.

Si tuviéramos programado el evento Refresh (no es este el caso) los atributos que allí aparecieran –y que no estuvieran dentro de un for each- también participarían de la determinación de la tabla base de esta parte fija.

También participarían los atributos del Tab Conditions, en este caso no hay ninguno. Es por ello que la tabla base será, entonces, Country.

Para confirmarlo veamos el listado de navegación. Si buscamos el Data Provider que corresponde entonces a la parte fija, vemos que efectivamente está navegando la tabla Country, quedándose con el país correspondiente recibido por parámetro.

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

```

Por otro lado, tenemos que ver cuál es la tabla base de la parte del grid. Por supuesto participarán los atributos pertenecientes al grid, en este caso SpeakerImage, SpeakerFullName, y SpeakerId.

También tenemos los atributos que se encuentran dentro del evento Load fuera de For each: y aquí tenemos al atributo SpeakerCVMini, que se usa varias veces en el evento. Pero además, tenemos el evento asociado a la imagen, y el evento asociado al Grid cuando hacemos Tap sobre el elemento, es el evento ViewSpeaker, veamos que atributos se usan en estos eventos.

En el evento Image1.tap tenemos al atributo SessionId, y en el evento View Speaker tenemos al atributo SpeakerId.

Hay otras cosas que debemos considerar, por ejemplo tenemos el Order de la grilla, acá tenemos la atributo SessionName que se usa para agrupar también los Oradores por Session.

Y tenemos los filtros en la propiedad Search, acá tenemos a los atributos SpeakerFullName, SessionName y SessionDescription.

Y por si esto fuera poco tenemos la propiedad Base Trn, donde indicamos Session.Speaker, también participan las Conditions a nivel del grid o a nivel del tab Conditions generales para todo el objeto, en ninguno de los dos tenemos ninguna condición y no usamos ningún atributo.

Resumiendo, tenemos atributos de la tabla Speaker (Id, Image, Full Name y Curriculum Vitae) y tenemos atributos de la tabla Session (Id, Name, Description). La tabla base entonces será SessionSpeaker ya que desde su tabla extendida obtenemos todos los atributos que necesitamos.

**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: CountryId, in: &SearchText, in: &start, in: &count, in: &grid, out: SessionSpeakersByCountry_Level_Detail_Grid1Std

**Warnings**

▲ sp0053 Conditional constraint "SpeakerFullName toupper() like "%\* + &SearchText.toupper() or SessionName.toupper() like "%\* + &SearchText.toupper() or SessionDescription.toupper() like "%\* + &SearchText.toupper()" cannot be generated in grid Grid1. Changed to standard constraint.

**LEVELS**

For Each SessionSpeaker (Line: 4)

Order: SessionId, SpeakerId  
 Index: CHARLACRADOR  
 Start from: FirstRecord  
 Loop while: NotEndOfTable  
 Navigation filters: 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

SessionSpeaker ( SessionId / SpeakerId ) INTO SessionId SpeakerId  
 Session ( SessionId ) INTO SessionName SessionDescription  
 Speaker ( SpeakerId ) INTO CountryId SpeakerFullName SpeakerImage Lit, SpeakerCVMini SpeakerImage SpeakerSurname SpeakerName

Si ahora vamos a observar la navegación del Data Provider correspondiente al grid, vemos que efectivamente está haciendo un for each a la tabla SessionSpeakers, y está filtrando por país. ¿Qué país? El recibido por parámetro.

Recordemos que las navegaciones de parte fija y grid son independientes. No se establecen automáticamente filtros entre una y otra. Lo que está haciendo que en la primera navegación, la de la parte fija y en el grid también se filtre por País, es el haber recibido el atributo CountryId como parámetro, y no el hecho de que las tablas base estén relacionadas.

## Web Panel WebSessionSpeakersByCountry Navigation Report

**Name:**  WebSessionSpeakersByCountry  
**Description:** Web Session Speakers By Country

**Environment:**  Default (C#)  
**Spec. Version:**  16\_0\_1-129648  
**Form Class:** HTML  
**Program Name:** WebSessionSpeakersByCountry  
**Parameters:** in: [CountryId](#)

## Event Load

**Order:** [SessionId](#) , [SpeakerId](#)  
**Navigation filters:** Index: ICHARLAORADOR  
 Start from: FirstRecord  
 Loop while: NotEndOfTable  
**Constraints:** [CountryId](#) = @[CountryId](#)  
**Join location:** Server

 [SessionSpeaker](#) ( [SessionId](#) , [SpeakerId](#) ) INTO [SessionId](#) [SpeakerId](#)  
 [Speaker](#) ( [SpeakerId](#) ) INTO [CountryId](#) [SpeakerCVMini](#) [SpeakerName](#) [SpeakerImage](#) [SpeakerImage.Uri](#)  
 [Country](#) ( [CountryId](#) ) INTO [CountryName](#)

A modo de ejemplo y para que queden claras las diferencias entre Web Panels y los objetos para Smart Devices, hice un panel similar pero en entorno Web.

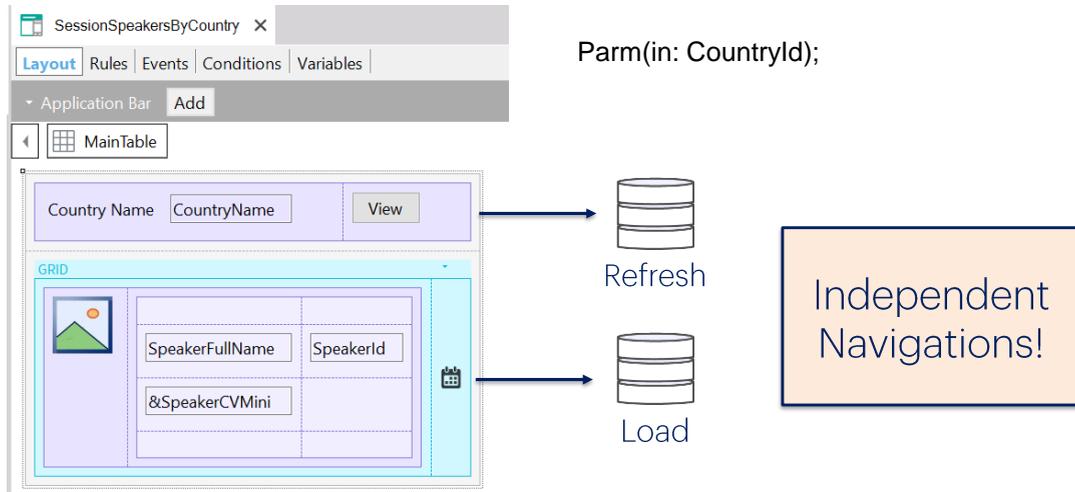
Tenemos el atributo CountryName en el form y la grilla con los mismos atributos, SpeakerName, SpeakerId, la variable, SpeakerImage, esta imagen, y los eventos son similares, usan los mismos atributos y llaman a objetos que resuelven de forma similar la misma funcionalidad.

Lo interesante es ver la navegación.

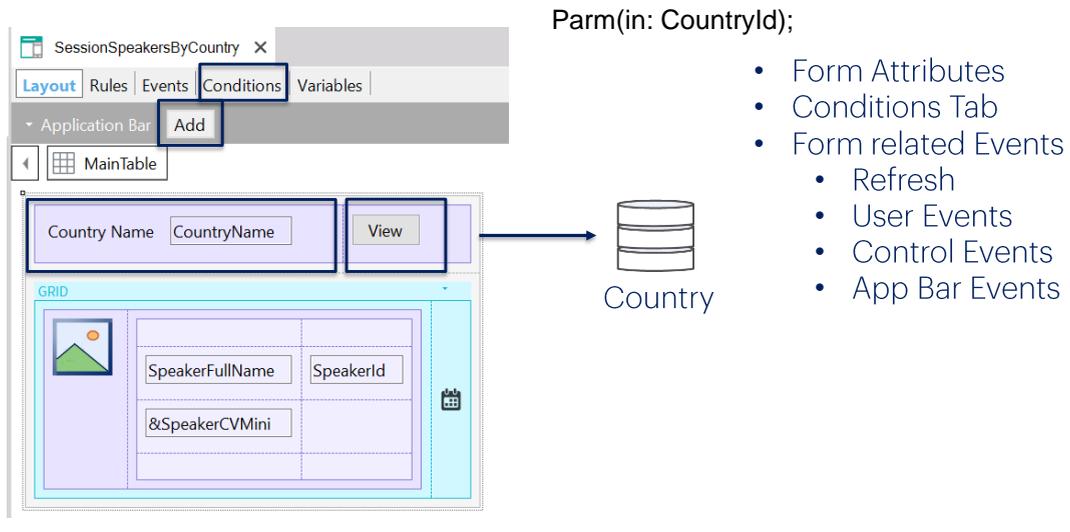
Primero notemos que los paneles tienen una sola navegación, si observamos la tabla base del evento Load es SessionSpeaker y desde esa tabla se va a resolver por medio de la tabla extendida el resto de los atributos necesarios.

volvamos a la presentación.

## Base Tables &amp; Navigation



Resumiendo lo visto en la demo: Parte fija y grid tienen cada uno su tabla base, y las navegaciones de cada una se determinan con independencia.

**Base Table: Fixed Part**

Los atributos que determinarán la tabla base de la parte fija son:

- Los atributos incluidos en la parte fija del layout, en nuestro caso CountryName
- Los atributos referenciados en las conditions generales (tab: conditions), en nuestro caso ninguna
- Los atributos que aparezcan en los eventos del form, fuera de comandos For each, esto es, los del Refresh y eventos asociados a controles de la parte fija del form, en nuestro caso el botón View, y del Application Bar (el botón Add).

Por tanto la tabla base en nuestro caso será COUNTRY.

**Base Table: Grid Part**

Orders	(1 order)
Search	(1 filter)
Conditions	

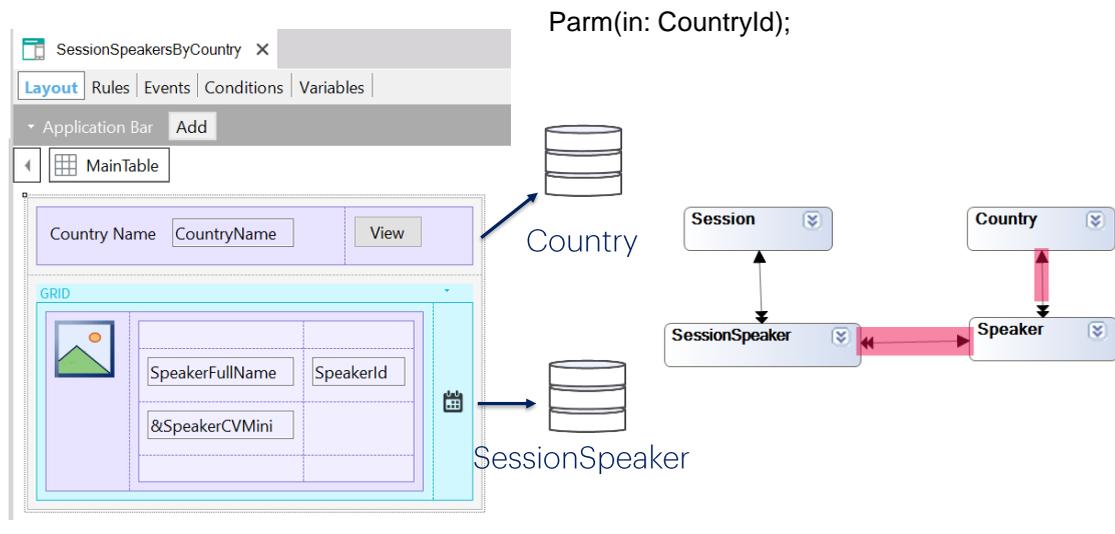
- Grid Attributes
- Conditions (Tab and Grid), Order, Search
- Grid related events
  - Load
  - Grid Events
  - User Grid Events

Los atributos que determinarán la tabla base del grid son:

- Los atributos incluidos en el grid
- Los atributos referenciados en el order, search, advanced search y conditions del grid, y las del tab Conditions, generales
- Los atributos que aparezcan en los eventos del grid, fuera de comandos for each. Esto es: los del Load y eventos asociados a controles dentro del grid (en nuestro caso, el del evento "ViewSession" y el Tap sobre la imagen).
- Si se configuró la propiedad Base Trn del grid, la tabla base surge de aquí, pero los atributos anteriores que mencionábamos, tienen que pertenecer a su extendida, para que esté todo bien definido.

Por todo esto, en nuestro ejemplo, la tabla base del grid será SessionSpeaker.

## Base Tables &amp; Navigation



Como ya sabemos, GeneXus identificó las tablas base de cada parte, Country y SessionSpeaker.

Estas tablas, si bien tienen una relación indirecta 1 a N, por defecto las navegaciones son independientes.

Esto significa que es como tener un par de For eachs paralelos (no anidados). Si queremos vincular las navegaciones, tenemos que hacerlo explícitamente (en este ejemplo, fue gracias a que recibimos en un parámetro atributo).

Observemos también que como la parte fija debe recuperar un único registro, para que el panel esté bien diseñado, necesitamos que se pueda filtrar la tabla base correspondiente (Country) para recuperar "el" registro deseado.

Como estamos recibiendo por parámetro en el atributo CountryId, esto se realizará automáticamente. Además, como el grid navega la tabla SessionSpeakers, que tiene a CountryId en su extendida, también se realizará sobre esta tabla el filtro, y sólo se mostrarán las sessions de speakers de ese país.

# GeneXus™

Videos	<a href="https://training.genexus.com">training.genexus.com</a>
Documentation	<a href="https://wiki.genexus.com">wiki.genexus.com</a>
Certifications	<a href="https://training.genexus.com/certifications">training.genexus.com/certifications</a>



Vamos a recordar rápidamente los conceptos de Tabla Base y Tabla Extendida.

Llamamos Tabla Base a cualquier tabla de la base de datos en donde estemos posicionados en un momento determinado y siempre que GeneXus deba hacer una operación sobre una tabla (recuperar datos, actualizarlos, eliminarlos) va a determinar una table base.

La tabla base siempre se va a tratar de una tabla de la base de datos, que generalmente es una tabla física pero podría ser una vista si estuviéramos hablando de transacciones dinámicas, en ese caso el tratamiento es el mismo.

Por otro lado la Tabla Extendida es una tabla virtual y siempre hablaremos de la tabla extendida de una tabla base determinada.

Esta tabla virtual estará compuesta por la table base y todas aquellas tablas con las cuales la tabla base tenga una relación N a 1, ya sea directa o indirectamente.