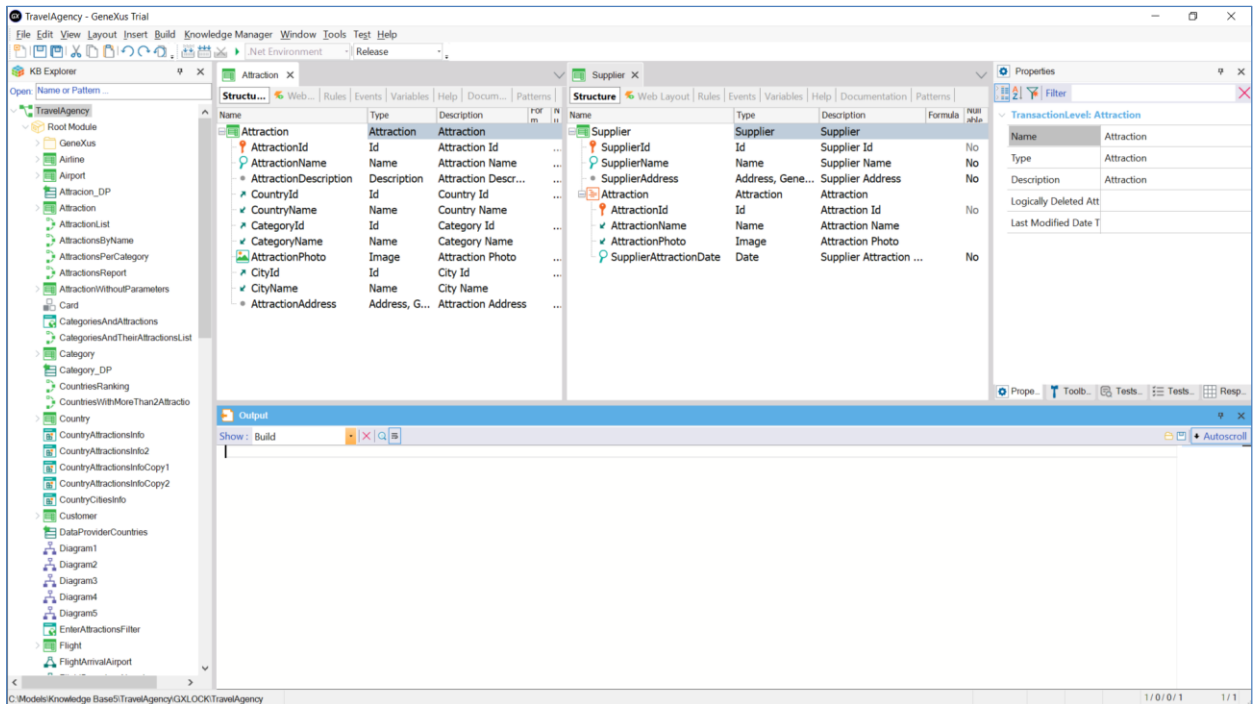


Algunas consideraciones sobre carga de grids sin tabla base

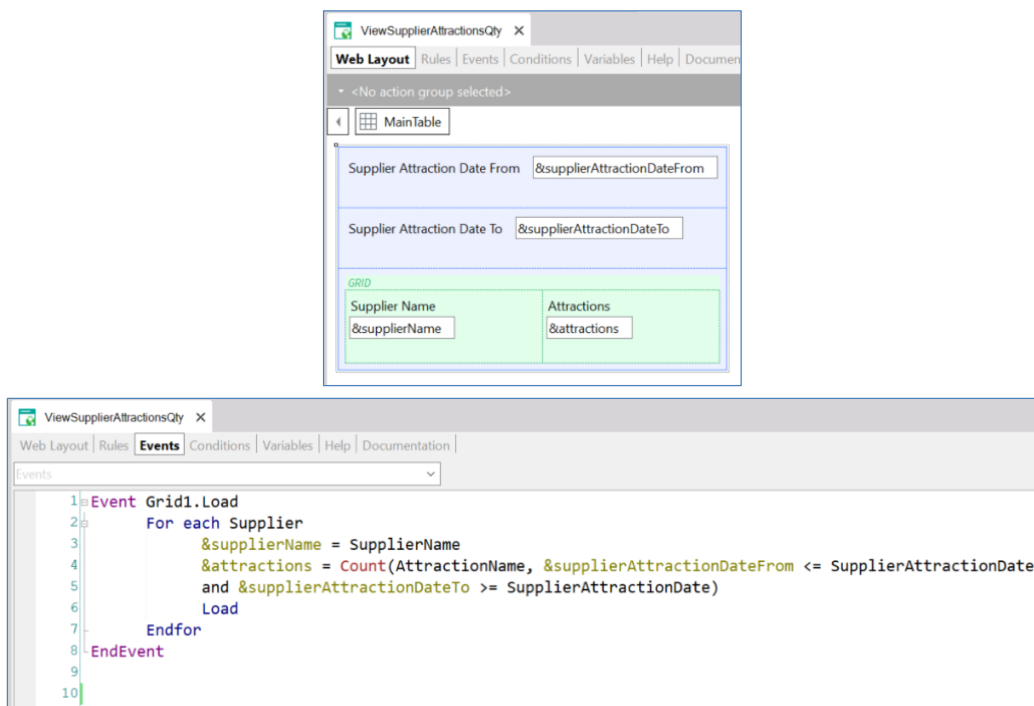
GeneXus™



Para la implementación de los diferentes ejemplos que veremos en este video, utilizaremos la aplicación para una agencia de viajes que venimos usando en videos anteriores.

Veamos que en nuestra aplicación tenemos entre otras, la transacción Attraction, y la transacción Supplier. Y esta última tiene como segundo nivel estos atributos de la transacción Attraction, más un nuevo atributo, SupplierAttractionDate, el cual se utilizará para registrar la fecha que el proveedor ofrece la atracción.

Por lo que cada Proveedor tendrá N atracciones.



Supongamos que tenemos el siguiente Web Panel.

Tenemos en su parte fija, dos variables del tipo Date, SupplierAttractionDateFrom y SupplierAttractionDateTo. Y tenemos un Grid, con dos variables dentro, SupplierName y attractions. La primera basada en el atributo SupplierName perteneciente a la tabla Supplier, y la otra del tipo numérico.

Lo que nos interesa implementar en este Web Panel será lo siguiente: Que el usuario pueda seleccionar dos fechas, desde y hasta. Y que sean mostrados en pantalla todos los nombres de los proveedores (Supplier), con su respectiva cantidad de atracciones. Teniendo en cuenta que para cada proveedor sólo nos interesan contar sus atracciones, cuya fecha esté comprendida entre las ingresadas por el usuario.

Veamos como lo implementamos en la sección Eventos.

En el evento Load del grid, tenemos un for each que recorrerá la tabla Supplier.

Para cada registro que encuentre se le asignará a la variable &SupplierName el valor del atributo SupplierName.

Y para la variable attractions, se definió una fórmula in line, que contará todas las atracciones del proveedor, con la condición de que la fecha de la atracción del proveedor (SupplierAttractionDate) en el que estamos parados, se encuentre dentro del período de fechas ingresado por el usuario.

Este es un web panel sin tabla base, ya que no existen atributos declarados en el Form, ni en las propiedades del grid, ni tampoco en los eventos existen atributos que estén fuera de un contexto. Este que vemos aquí se encuentran dentro del for each, por lo que, en tal caso, si no tuviéramos esta transacción ingresada, ayudaría a determinar la tabla que recorrerá el for each, pero no influye en la determinación de la tabla base del grid. Y estos otros atributos también están dentro de un contexto, el de la fórmula Count. Sólo ayudaran a determinar la tabla a ser navegada por esta fórmula.

Es por esto que debemos ingresar el comando Load, para cargar en el grid cada registro encontrado.

Tal como está programado, podríamos pensar que es necesario agregar un botón en nuestro panel, para que luego de ingresadas las fechas, hagamos clic en él y actualice nuestra grilla.

En ejecución

Travel Agency

Recents View Supplier Attr...

Supplier Attraction Date From 03/08/21

Supplier Attraction Date To 04/30/21

Supplier Name	Attractions
Transport Travel	5
TipGroup Operator	6
Trafalgar	2
Intrepid Travel	3

Probemos ejecutar tal como está.

Vemos que se carga el listado de todos los proveedores, y con cantidad de atracciones 0, debido a que todavía no ingresamos las fechas que condicionan la fórmula que las contará.

Ingresamos dos fechas, desde y hasta, que como recién vimos condicionaran al conteo de atracciones de cada proveedor. Y automáticamente se actualizará la cantidad de atracciones para cada proveedor condicionadas por las fechas ingresadas.

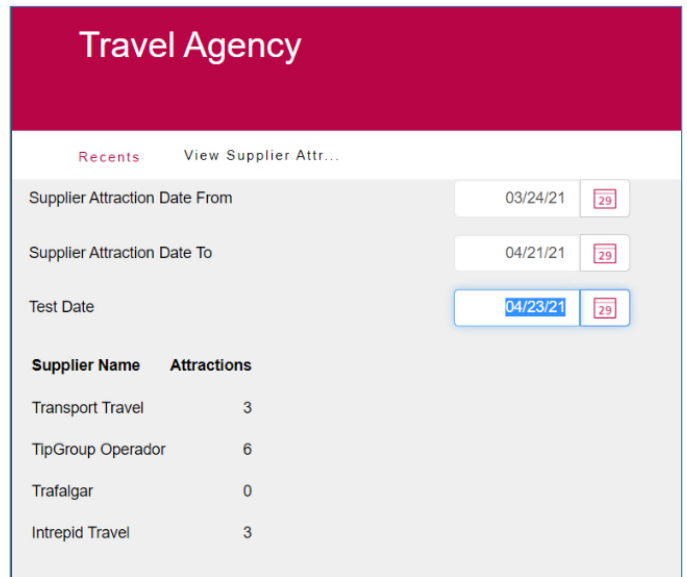
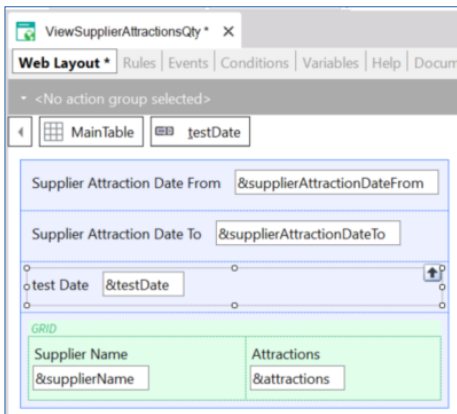
No fue necesario indicarle nosotros que actualice la grilla al cambiar las fechas, se hizo automáticamente. ¿Por qué tuvo este comportamiento?

The screenshot displays the GeneXus IDE interface. On the left, the 'Web Layout' editor shows a 'MainTable' containing two date filters: 'Supplier Attraction Date From' and 'Supplier Attraction Date To', both using the variable '&supplierAttractionDateFrom' and '&supplierAttractionDateTo' respectively. Below these is a 'GRID' with two columns: 'Supplier Name' (using variable '&supplierName') and 'Attractions' (using variable '&attractions').

On the right, the 'Properties' panel is open for the 'Web Panel: ViewSupplierAttractionsQty'. The 'Automatic refresh' property is highlighted with an orange border, showing its value as 'Yes'. Other visible properties include Name, Description, Module/Folder, Theme, Type, Master Page, Show Master Page w, Main program, On session timeout, Focus control, Cache expiration laps, Auto compress http, Qualified Name, and Object Visibility.

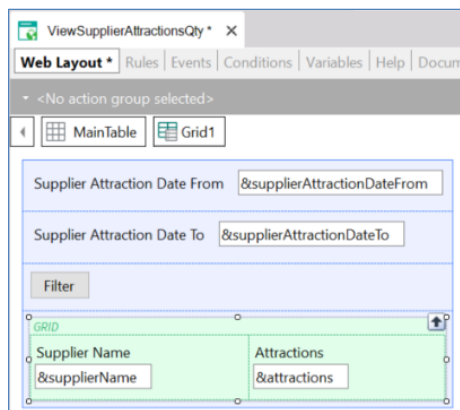
Si vamos a las propiedades de nuestro web panel, vemos una de nombre Automatic refresh, que por defecto está en Yes. Lo que hace esta propiedad, es realizar una actualización de la grilla cuando detecta cambios en alguna variable definida en el web layout y que es utilizada luego: en el evento Refresh, el evento Load del grid, o en las condiciones del grid o del web panel.

Es lo que está sucediendo en este caso, las variables SupplierAttractionDateFrom y SupplierAttractionDateTo, ingresadas para filtrar la cantidad de atracciones a mostrar. Son definidas luego dentro del evento Load del grid, más precisamente dentro de esta fórmula Count. Al encontrarse dentro de este evento, se entiende que estas variables tienen un impacto en los datos que estamos queriendo mostrar, por lo que cada vez que es modificada una de ellas, se actualiza nuestra grilla, o sea se dispara el evento Refresh del grid.



Veamos que pasa si agregamos una nueva variable a la parte fija de nuestro web panel, pero que luego no se encuentre definida en un evento Load, ni en un evento Refresh ni en ninguna conditions.

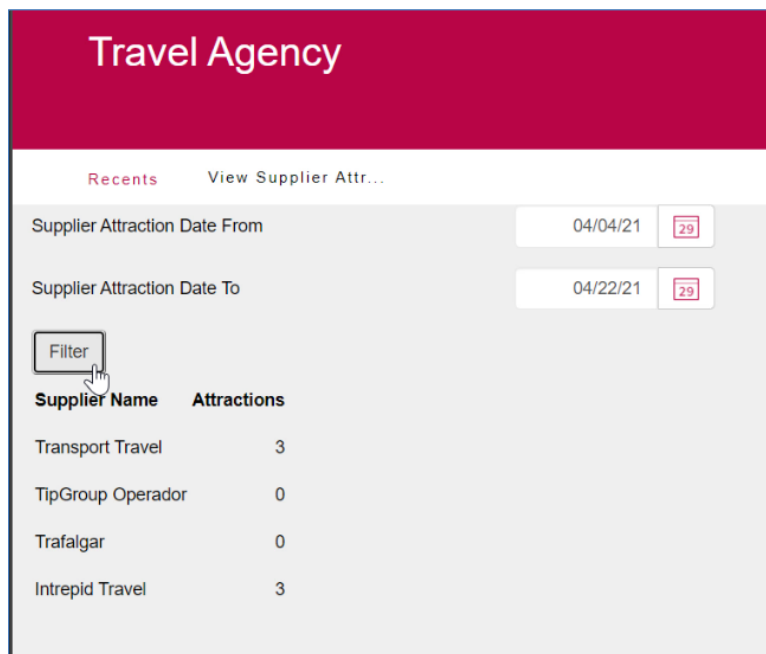
Vemos que al modificar el valor de esta variable no se dispara el evento refresh del grid, no hay ningún intento de actualizar la grilla, ya que como dijimos, para estos casos no aplica la propiedad Automatic refresh.



```

Event 'Filter'
  Grid1.Refresh()
Endevent

```



Quitemos esta variable.

Ahora supongamos que no queremos el comportamiento que nos da la propiedad Automatic refresh, ya que nos interesa que se actualice la grilla cuando nosotros queramos, y no cada vez que es modificada alguna de estas variables.

La opción sería dejar la propiedad Automatic refresh en No, y agregar un botón en la parte fija de nuestro panel que nos permita controlar cuando disparar esa actualización de la grilla.

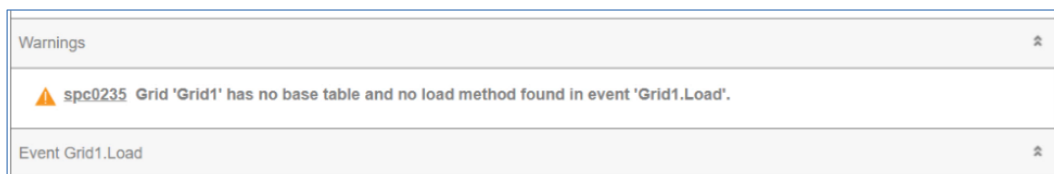
En el evento del botón simplemente definimos el método refresh de nuestro grid.

Vemos que ahora al cambiar los valores de estas variables ya no se actualiza nuestro grid. Al hacer clic al botón, se dispara el evento refresh del grid, y ahí sí, la grilla se actualizará.


```

1 Event Grid1.Load
2   For each Supplier
3     &supplierName = SupplierName
4     &attractions = Count(AttractionName, &supplierAttractionDateFrom <= SupplierAttractionDate
5     and &supplierAttractionDateTo >= SupplierAttractionDate)
6   Endfor
7 EndEvent
8

```



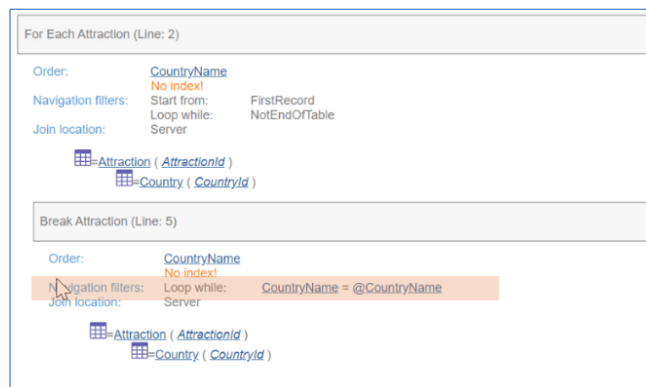
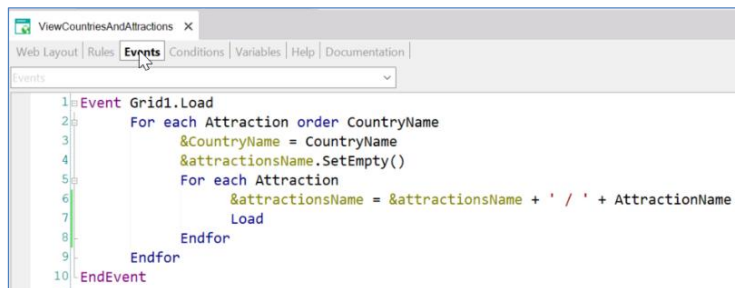
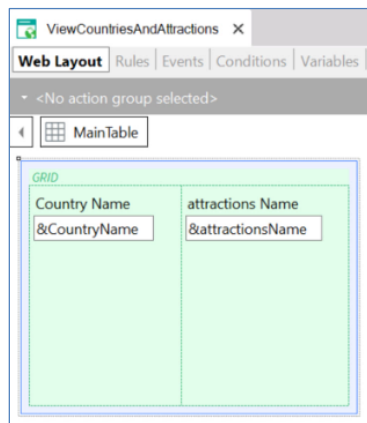
Quitemos el botón junto a su evento, y volvamos a dejar esta propiedad en Yes.

Supongamos ahora, que nos hubiésemos olvidado de declarar este comando Load dentro del evento. ¿Qué pasaría? ¿Cuál será el comportamiento?
Guardemos e implementemos este cambio.

Veamos el listado de navegación.

Nos aparece un Warning, indicando que nuestro grid no tiene tabla base y no tenemos declarado dentro del evento Load ningún comando Load. Este Warning también nos aparece en el output.

Al probarlo en ejecución, vemos que no se carga ningún registro en la grilla, ya que al omitir el comando Load, se interpreta que no nos interesa cargar ningún registro. Al ser esta una implementación que no tendría demasiado sentido, es que se dispara el warning a modo de advertencia.



Supongamos ahora, que, en esta aplicación, interesa tener un Web Panel que muestre todos los países que tengan al menos una atracción, y a continuación todas las atracciones, una al lado de otra, separadas por una barra.

Recordemos que nuestra transacción Attraction tiene al atributo CountryId y CountryName de la transacción Country. Por lo que cada atracción pertenecerá a un país. Para esto en el Web Layout tenemos un grid con dos variables, una basada en el atributo CountryName que mostrará el nombre de los países, y la otra del tipo LongVarChar que mostrará todas las atracciones.

En la sección eventos tenemos definido el evento Load del grid, y dentro dos for each anidados.

El externo recorrerá la tabla de atracciones ordenando los registros por nombre de país. Y le asigna el valor del atributo CountryName a la variable de dicho nombre.

Luego tenemos un for each anidado que navegará la misma tabla, Attraction. Y lo que se hará con esta definición, será concatenar todas las atracciones encontradas en una sola variable, attractionName.

Como sabemos, al for each interno recorrer la misma tabla que el externo, y habiendo definido un orden en este, se hará un corte de control. Y el criterio de corte lo determinará justamente el atributo declarado en la cláusula order del for each externo, o sea se agrupará por CountryName.

Lo vemos esto claramente en el listado de navegación.

En ejecución

Travel Agency	
Country Name	attractions Name
Brazil	/ Christ the Redemmer
Brazil	/ Christ the Redemmer / Christ the Redemmer
China	/ The Great Wall
China	/ The Great Wall / Meet the Emperor
China	/ The Great Wall / Meet the Emperor / The Great Wall
China	/ The Great Wall / Meet the Emperor / The Great Wall / Forbidden city
China	/ The Great Wall / Meet the Emperor / The Great Wall / Forbidden city / Forbidden city
China	/ The Great Wall / Meet the Emperor / The Great Wall / Forbidden city / Forbidden city / Meet the Emperor
England	/ London Towers
England	/ London Towers / London Towers
France	/ Eiffel Tower
France	/ Eiffel Tower / Louvre Museum
France	/ Eiffel Tower / Louvre Museum / Matisse Museum

Ejecutemos para probar.

No estamos obteniendo el resultado deseado.

Por ejemplo China, se está imprimiendo una vez junto a su primera atracción. Luego se imprime nuevamente con sus primeras dos atracciones concatenadas, luego se imprime nuevamente con sus primeras tres atracciones concatenadas, etc. Y esto mismo pasa con todos los países.

¿Qué es lo que está sucediendo? ¿Cuál fue el error en nuestra definición?

```

Grid1.Load
1 Event Grid1.Load
2 For each Attraction order CountryName
3   &CountryName = CountryName
4   &attractionsName.SetEmpty()
5   For each Attraction
6     &attractionsName = &attractionsName + ' / ' + AttractionName
7   Endfor
8   Load
9 Endfor
10 EndEvent

```

Travel Agency

Recents View Countries And...

Country Name	attractions Name
Brazil	/ Christ the Redemmer
China	/ The Great Wall / Forbidden city / Meet the Emperor
England	/ London Towers
France	/ Louvre Museum / Matisse Museum / Eiffel Tower
Italy	/ Cinque Terre / Rifugio Nuvolau
Scotland	/ Glenfinnan Viaduct
United States	/ Smithsonian Institute / London Bridges

El error aquí está en donde declaramos el comando Load. Lo hicimos dentro del for each anidado. Por lo que cada vez que encuentra una atracción, y la concatena a nuestra variable lo imprime en pantalla, y esto no es lo que deseamos. Queremos que una vez sean concatenadas todas las atracciones de ese país, recién ahí se muestre esta información. Por lo que tenemos que definir el comando Load fuera de este for each.

Hagámoslo.
Volvamos a ejecutar.

Vemos que ahora sí, tiene el funcionamiento esperado.
Para indagar y profundizar sobre los temas tratados en este video, los invitamos a ingresar en nuestro Wiki.