

Pantallas web

Tabla base y navegación en Web panel con UN grid

GeneXus™

Web Panel con un Grid

Ahora pasemos a ver el caso de web panel con un grid.

The image displays a web application interface for "View Country Info Simple". The interface is divided into two main sections: a design view on the left and a live application view on the right.

Design View (Left): Titled "With one Grid", it shows a web form with a "Rules" tab selected. A rule is defined as `1 | parm(in: CountryId);`. The form layout includes a "Country Name" field, two input fields for "Attraction Name From" and "Attraction Name To", a "GRID" section with columns for "Attraction Id", "Attraction Name", "Attraction Photo", and "Trips", and a "Total Trips" field.

Live Application View (Right): Titled "Application Name", it shows the rendered form. The "COUNTRY NAME" is "FRANCE". The "Attraction Name From" and "Attraction Name To" fields are empty. The "GRID" displays a table of attractions:

Attraction Name	Attraction Photo	Trips
Louvre Museum		0 Update New trip
Eiffel Tower		0 Update New trip
Matisse Museum		0 Update New trip

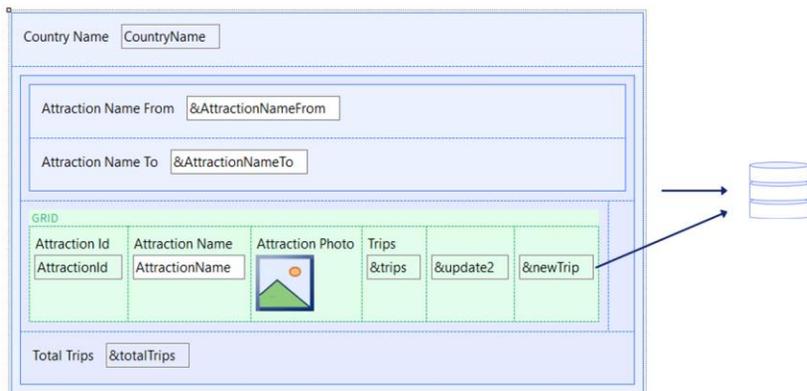
Below the grid, the "Total Trips" is displayed as 0.

Aquí estamos recibiendo por parámetro el identificador de un país, y mostrando en un grid todas sus atracciones turísticas, y en la parte fija el nombre de ese país.

With one Grid

Web Form **Rules** Events Conditions Variables

```
1 param( in: CountryId );
```

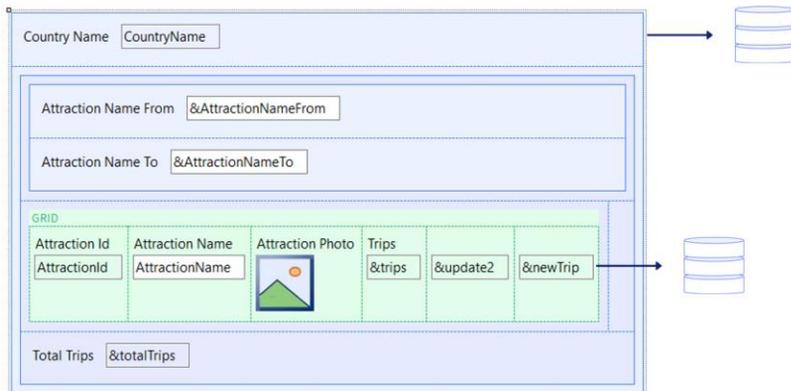


Cuando el web panel tiene a lo sumo un grid, podemos decir que de existir tabla base, es indistinto hablar de la tabla base del web panel que de la tabla base del grid. ¿Por qué?

With one Grid

Web Form **Rules** Events Conditions Variables

```
1 param( in: CountryId );
```



Porque no podrá suceder, como sí ocurre en el caso de los Panels a secas, es decir, los desarrollados para ejecutar con Angular, por ejemplo o con Android o Apple, que exista una navegación para la parte fija del panel y otra diferente para el grid.

With one Grid

Web Form **Rules** Events Conditions Variables

```
1 param( in: CountryId );
```

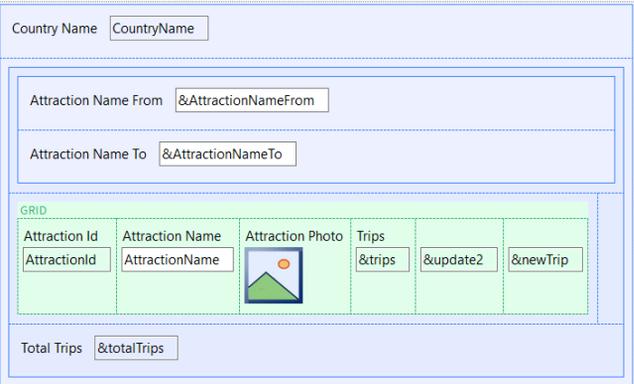
The screenshot shows a web form design interface. At the top, there are tabs for 'Web Form', 'Rules', 'Events', 'Conditions', and 'Variables'. Below the tabs, there is a code editor with the text '1 param(in: CountryId);'. The main area displays a form layout. At the top, there is a 'Country Name' label and a text input field containing 'CountryName'. Below this, there are two more text input fields: 'Attraction Name From' with the value '&AttractionNameFrom' and 'Attraction Name To' with the value '&AttractionNameTo'. A large orange circle highlights the 'CountryName' input field, and a blue arrow points from it to a database icon on the right. Below these fields is a 'GRID' section with a light green background. The grid has six columns: 'Attraction Id' (with 'AttractionId' below), 'Attraction Name' (with 'AttractionName' below), 'Attraction Photo' (with a photo icon below), 'Trips' (with '&trips' below), '&update2', and '&newTrip'. A blue arrow points from the '&newTrip' cell to the database icon. At the bottom of the grid area, there is a 'Total Trips' label and a text input field containing '&totalTrips'. At the very bottom of the screen, there are several small navigation icons.

Para los web panels con un grid, de haber tabla base, será única, y lo que deberá suceder si está bien implementado es que los atributos de la parte fija deberán tener el mismo valor para todos los registros de la tabla base que se vayan a cargar en el grid, porque de lo contrario ¡deberían estar en el grid y no en la parte fija!

With one Grid

Web Form **Rules** Events Conditions Variables

```
1= parm( in: CountryId );
```



Attraction Id	Attraction Name	Attraction Photo	Trips	Total Trips
AttractionId	AttractionName		&trips &update2 &newTrip	&totalTrips

Grid1's Conditions

```
AttractionName >= &AttractionNameFrom
when not &AttractionNameFrom.IsEmpty();

AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();
```

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName
Conditions	AttractionName >= &AttractionNameFrom...
Unique	
Save State	False
Data Selector	(none)

Aquí se cargarán en el grid todas las atracciones que pasen los filtros de las conditions y de pertenecer al país recibido por parámetro. Por lo tanto, para todas ellas, el CountryName será el mismo y por eso está en la parte fija.

Observemos que estaremos recibiendo por parámetro en el atributo CountryId que NO PARTICIPA en la determinación de la posible tabla base, pero sí actúa como filtro sobre las navegaciones que GeneXus encuentre, tal como vimos en el caso anterior.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

Country Name

Attraction Name From

Attraction Name To

GRID					
Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
AttractionId	AttractionName		&trips	&update2	&newTrip

Total Trips

- Attributes in the **form** (visible or hidden)
- **Grid** Base Trn property
- **Grid** Order property
- **Grid** Conditions property
- Grid Unique property
- **Grid** Data Selector property
- Attributes in **events** (without context, ie: For each command and inline aggregate formula)

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName
Conditions	AttractionName >= &AttractionNameFrom...
Unique	
Save State	False
Data Selector	(none)

Grid1's Conditions

```
AttractionName >= &AttractionNameFrom
when not &AttractionNameFrom.IsEmpty();

AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();
```

extended table



¿Qué partes del web panel son tenidas en cuenta para determinar tabla base? Evidentemente, al aparecer un grid, además de los lugares que ya habíamos visto para el caso anterior, se suma todo lo relativo al grid.

Por tanto, si allí aparecen atributos, o en las propiedades Order, conditions, Unique, Data Selector y, ni hablar, si aparece especificada transacción base en la propiedad correspondiente, entonces el grid (y por tanto el Web panel) tendrá una tabla base implícita.

Por supuesto, de existir transacción base, deberá cumplirse que todos los atributos del form y los que estén en eventos “sueños” deberán pertenecer a su tabla extendida. De lo contrario GeneXus nos advertirá cuáles atributos no podrán ser “alcanzables” y quedarán sin valor.

Si no se especificó transacción base, entonces, como en el caso del web panel sin grid, GeneXus la encontrará buscando la mínima tabla extendida que contenga a todos los atributos que aparezcan en los lugares que definimos. Su tabla base será la tabla base elegida.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

Country Name

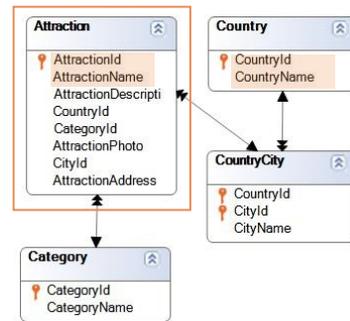
Attraction Name From

Attraction Name To

GRID

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
AttractionId	AttractionName		&trips	&update2	&newTrip

Total Trips



Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName
Conditions	AttractionName >= &AttractionNameFrom...
Unique	Grid's Conditions
Save State	False
Data Selector	(none)

Grid's Conditions

```

AttractionName >= &AttractionNameFrom
when not &AttractionNameFrom.IsEmpty();

AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();
  
```

Antes de pasar a observar los eventos, ya por existir transacción base podemos decir que la tabla base será Attraction. Y corroboramos que todos estos otros atributos pertenecen a su tabla extendida. Por lo que, si no hubiera ni un solo evento programado, igual sabríamos que se va a recorrer la tabla Attraction, ordenándola por CountryId y AttractionName, filtrando por las atracciones cuyo AttractionName cumpla con estas dos condiciones, y además cuyo país corresponda al recibido por parámetro.

Digámoslo una última vez: para los web panels si hay grid, los atributos de la parte fija corresponden a la navegación del grid. Luego veremos que si hay varios grids, corresponderán a la navegación de uno de ellos.

Si no estuviéramos recibiendo en el atributo CountryId, entonces, dado que se cargarán atracciones de distintos países en el grid, ¿qué valor asumirá el atributo CountryName en pantalla? Es fácil sospechar que será o el primero o el último de los registros cargado en el grid.

Probémoslo.

With one Grid

Web Panel ViewCountryInfoSimple Navigation Report

Name: ViewCountryInfoSimple
Description: View Country Info Simple

Environment: Default (C#)
Spec. Version: 17_0_0-144971
Form Class: HTML
Program Name: ViewCountryInfoSimple
Parameters: in: CountryId

Warnings

spec0038 There is no index for order CountryId, AttractionName: poor performance may be noticed in grid 'Grid1'.

Event Grid1.Load

```

Order:      CountryId, AttractionName
            No index!
Navigation filters:
Start from: CountryId = @CountryId
Loop while: CountryId = @CountryId
Constraints: AttractionName >= &AttractionNameFrom WHEN not &AttractionNameFrom.isempty()
              AttractionName <= &AttractionNameTo WHEN not &AttractionNameTo.isempty()
Join location: Server
            
```

```

Attraction (AttractionId)
├── Country (CountryId)
│   └── count (TripDate) navigation
│       ├── TripAttraction (AttractionId)
│       └── Trip (TripId)
            
```

GeneXus Application Name

COUNTRY NAME: FRANCE

Attraction Name From:

Attraction Name To:

Attraction Name	Attraction Photo	Trips		
Louvre Museum		0	Update	New trip
Lifel Lower		0	Update	New trip
Matisse Museum		0	Update	New trip

Total Trips: 0

Aquí tenemos el web panel, correctamente programado. Veamos, antes, su listado de navegación.

Vemos que, efectivamente, hay una tabla base, que es Attraction. La vemos claramente identificada a nivel del evento Load del grid.

Y ahora vamos a ejecutar antes de modificar el parámetro. Desde este otro web panel invocamos al nuestro.

With one Grid

Web Panel ViewCountryInfoSimple Navigation Report

Name: ViewCountryInfoSimple
Description: View Country Info Simple

Environment: Default (C#)
Spec. Version: 17_0_0-144971
Form Class: HTML
Program Name: ViewCountryInfoSimple
Parameters: in: &CountryId

Warnings

▲ spc0038 There is no index for order CountryId, AttractionName: poor performance may be noticed in grid 'Grid1'.

Event Grid1 Load

Order: CountryId, AttractionName
 No index!

Navigation filters: Start from: FirstRecord
 Loop while: NotEndOfTable

Constraints: AttractionName >= &AttractionNameFrom WHEN not &AttractionNameFrom.isempty()
 AttractionName <= &AttractionNameTo WHEN not &AttractionNameTo.isempty()

Join location: Server

```

  [Attraction (AttractionId)]
  [Country (CountryId)]
  [count( TripDate ) navigation]
  [TripAttraction (AttractionId)]
  [Trip (TripId)]
  
```

GeneXus Application Name

COUNTRY NAME: BRAZIL

Attraction Name From:

Attraction Name To:

Attraction Name	Attraction Photo	Trips	Update	New trip
Louvre Museum		0	Update	New trip
The Great Wall		0	Update	New trip
Eifel Tower		0	Update	New trip
Christ the Redeemer		0	Update	New trip
Smithsonian Institute		0	Update	New trip
Matisse Museum		0	Update	New trip
Forbidden city		0	Update	New trip
Glenfiddich Viaduct		0	Update	New trip
Meet the Emperor		0	Update	New trip

Ahora modifiquemos nuestro panel para que en lugar de recibir en el atributo CountryId, se reciba en la variable &CountryId. Y de este modo no haya un filtro automático.

Observemos el listado de navegación. Vemos que ahora no habrá filtro.

Y vamos a verlo en ejecución. Vemos que cargó como nombre de país el del primero de los registros, no el del último.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

Country Name

Attraction Name From

Attraction Name To

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
AttractionId	AttractionName		&trips	&update2	&newTrip

Total Trips

```

Event Grid1.Load
    &trips = Count(TripDate)
    &totalTrips = &totalTrips + &trips
Endevent

Event Grid1.Refresh
    &totalTrips = 0
Endevent

Event Start
    &newTrip = "New trip"
    &update2 = "UPDATE"
    CountryName.ForeColor = RGB(147,4,55) //DarkBase
    CountryName.FontBold = True
Endevent

Event &newTrip.Click
    &trips = NewTrip(AttractionId)
    Refresh
endevent

Event AttractionName.Click
    ViewAttractionFromScratch(AttractionId)
Endevent

```

Bien, dejemos el parámetro como lo teníamos.

Si ahora vayamos a observar los eventos, reparamos en el hecho de que evidentemente el evento Load se programó ya sabiendo que el grid tendría tabla base. Si analizamos como hace GeneXus qué atributos participarán en la determinación de la tabla base, vemos que son todos estos que están "sueños".

Claramente TripDate no. Este solo participa en la determinación de la tabla a ser navegada por la fórmula count. Por supuesto, y al igual que ocurre con el web panel en general, si el grid tiene tabla base entonces esta fórmula count además podrá tener una condición implícita, que en este caso es que contará los trips de la atracción turística que se está por cargar.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

Country Name

Attraction Name From

Attraction Name To

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
<input type="text" value="&AttractionId"/>	<input type="text" value="&AttractionName"/>		<input type="text" value="&trips"/>	<input type="text" value="&update2"/>	<input type="text" value="&newTrip"/>

Total Trips

```

Event Grid1.Load
  For each
    &CountryName = CountryName
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &AttractionPhoto = AttractionPhoto
    &trips = Count(TripDate)
    Load
    &totalTrips = &totalTrips + &trips
  endfor
Endevent

Event Grid1.Refresh
  &totalTrips = 0
Endevent

Event Start
  &newTrip = "New trip"
  &update2 = "UPDATE"
  &CountryName.ForeColor = RGB(147,4,55) //DarkBase
  &CountryName.FontBold = True
Endevent

Event &newTrip.Click
  &trips = NewTrip(&AttractionId)
  Refresh
endevent

Event &AttractionName.Click
  ViewAttractionFromScratch(&AttractionId)
Endevent

```

Como vimos antes, no alcanza con que sustituyamos todos los atributos que mostramos por variables en todos lados si olvidamos hacerlo en uno de estos lugares. Por ejemplo, aquí hicimos un Save as del panel anterior, para implementarlo sin tabla base.

Cambiamos todos los atributos del form por variables, y también esos mismos atributos en los eventos.

De hecho programamos el evento Load del grid especificando el For each para recorrer la tabla Attraction y cargar todas las variables del grid (y también la de la parte fija, &CountryName), pensando que estamos programando el web panel sin tabla base.

No le especificamos transacción base al For each, pero podríamos haberlo hecho. Sabemos que se va a filtrar por CountryId porque se está recibiendo por parámetro y además que ese atributo de la regla parm no participa en determinar tabla base.

Pero, ¿dónde estamos especificando los filtros por nombre de atracción? Los olvidamos. ¡Se nos pasó por alto modificar también las propiedades del grid! Y las propiedades del grid sí determinan tabla base.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

Country Name

Attraction Name From

Attraction Name To

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
&AttractionId	&AttractionName		&trips	&update2	&newTrip

Total Trips

```
Event Grid1.Load
For each
  &CountryName = CountryName
  &AttractionId = AttractionId
  &AttractionName = AttractionName
  &AttractionPhoto = AttractionPhoto
  &trips = Count(TripDate)
  Load
  &totalTrips = &totalTrips + &trips
endfor
Endevent
```

Event Grid1.Refresh

Grid: Grid1

Control Name	Grid1
Collection	
Base Trn	Attraction
Order	CountryId, AttractionName
Conditions	AttractionName >= &AttractionNameFrom... //DarkBase
Unique	
Save State	False
Data Selector	(none)

```
endevent
AttractionName <= &AttractionNameTo
when not &AttractionNameTo.IsEmpty();

Event &AttractionN
ViewAttractionFromScratch(&AttractionId)
Endevent
```

Si observamos, estas tres propiedades que omitimos, ESTÁN ESTABLECIENDO TABLA BASE.

With one Grid

Warnings

▲ spc0038 There is no index for order CountryId, AttractionName: poor performance may be noticed in group starting at line 2.

Event Grid1.Load

Order: CountryId, AttractionName
No index

Navigation filters: Start from: CountryId = @CountryId
Loop while: CountryId = @CountryId

Constraints: AttractionName >= &AttractionNameFrom WHEN not &AttractionNameFrom.isempty()
AttractionName <= &AttractionNameTo WHEN not &AttractionNameTo.isempty()

Join location: Server

Attraction (AttractionId)
Country (CountryId)

Break Attraction (Line: 2)

Order: CountryId, AttractionName
No index

Navigation filters: Loop while: CountryId = @CountryId and AttractionName = @AttractionName

Join location: Server

Attraction (AttractionId)
Country (CountryId)
count(TripDate).navigation
TripAttraction (AttractionId)
Trip (TripId)



```

Event Grid1.Load
For each
    &CountryName = CountryName
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &AttractionPhoto = AttractionPhoto
    &trips = Count(TripDate)
    Load
    &totalTrips = &totalTrips + &trips
endfor
Endevent

Event Grid1.Refresh
&totalTrips = 0
Endevent

Event Start
&newTrip = "New trip"
&update2 = "UPDATE"
&CountryName.ForeColor = RGB(147,4,55) //DarkBase
&CountryName.FontBold = True
Endevent

Event &newTrip.Click
&trips = NewTrip(&AttractionId)
Refresh
endevent

Event &AttractionName.Click
ViewAttractionFromScratch(&AttractionId)
Endevent

```

Observemos, de hecho, qué nos dice el listado de navegación. Justamente, encontré tabla base Attraction, por lo que ejecutará el Load N veces, filtrando por país recibido por parámetro y también por las variables de acuerdo a las conditions que especificamos en el grid. Y para cada registro de esos, ejecutará el For each del Load, realizando un corte de control, porque la tabla base es la misma, Attraction.

¡Todo por olvidar revisar las propiedades del grid!

With one Grid

```

Event Grid1.Load
  For each Attraction
    order CountryId, AttractionName
    where AttractionName >= &AttractionNameFrom when not &AttractionNameFrom.IsEmpty()
    where AttractionName <= &AttractionNameTo when not &AttractionNameTo.IsEmpty()
    &CountryName = CountryName
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &AttractionPhoto = AttractionPhoto
    &trips = Count(TripDate)
    Load
    &totalTrips = &totalTrips + &trips
  endFor
Endevent

```

Environment: Default (C#)
 Spec. Version: 17_0_0-144971
 Form Class: HTML
 Program Name: ViewCountryInfoSimple2
 Parameters: in: [CountryId](#)

Properties

General	Class
Control Name	Grid1
Collection	
Base Tm	
Order	
Conditions	
Unique	
Save State	False
Data Selector	(none)

Event Grid1.Load

For Each Attraction (Line: 2)

Order: [CountryId](#), [AttractionName](#)
 No index!

Navigation filters: Start from: [CountryId](#) = @CountryId
 Loop while: [CountryId](#) = @CountryId
 Constraints: [AttractionName](#) = &AttractionNameFrom WHEN not &AttractionNameFrom.IsEmpty()
 Join location: [AttractionName](#) <= &AttractionNameTo WHEN not &AttractionNameTo.IsEmpty()
 Server

Attraction (AttractionId)
 Country (CountryId)
 count(TripDate).navigation
 TripAttraction (AttractionId)
 Trip (TripId)

Así que lo que debemos hacer es llevar estas propiedades al For each.

Primero la transacción base...

Luego el order...

Y por último las conditions como cláusulas where...

Grabamos. Nos está dando este error porque no lleva punto y coma en la sintaxis del for each. Volvamos a grabar. Ahora sí.

Y veamos ahora el listado de navegación.

Ahora sí, conseguimos que el web panel no tuviera tabla base.

Veamos que si colocáramos las cláusulas where como Conditions pero no del grid, sino como conditions generales del Web panel, al igual que sucedía en el caso de un panel sin grid, los atributos de estas condiciones no participan en determinar tabla base. Aquí lo vemos claramente. El listado de navegación no se diferencia del anterior.

El problema de las conditions generales es que van a aplicar como filtros en toda la navegación del web panel que involucre a sus atributos, y a veces eso no nos interesa. Pensemos, por ejemplo, en el caso de tener múltiples grids.

With one Grid

```

Event Grid1.Load
  For each Attraction
    order CountryId, AttractionName
    &CountryName = CountryName
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &AttractionPhoto = AttractionPhoto
    &trips = Count(TripDate)
    Load
    &totalTrips = &totalTrips + &trips
  endfor
Endevent

```

Only load
an
attraction
if &trips > 1
?

```

Event Grid1.Load
  &trips = Count(TripDate)
  &totalTrips = &totalTrips + &trips
Endevent

```

En general es mejor práctica implementar los web panels con tabla base, puesto que nos ahorra todo este trabajo. Aquí tuvimos que establecer el for each, y asignarle a las variables los valores de los atributos y especificar el comando Load para cargar cada línea. En cambio en la versión con tabla base vemos que todo fue tanto más sencillo.

Pero, ¿cómo haríamos si quisiéramos solo cargar atracciones que estén en más de una excursión (trip)?

With one Grid

```

Event Grid1.Load
  For each Attraction
    order CountryId, AttractionName
    &CountryName = CountryName
    &AttractionId = AttractionId
    &AttractionName = AttractionName
    &AttractionPhoto = AttractionPhoto
    &trips = Count(TripDate)
    if &trips > 1
      Load
    endif
    &totalTrips = &totalTrips + &trips
  endfor
Endevent

```

Only load
an
attraction
if &trips > 1
?

```

Event Grid1.Load
  &trips = Count(TripDate)
  If &trips > 1
    Load
  endif
  &totalTrips = &totalTrips + &trips
Endevent

```

Country Name:

Attraction Name From:

Attraction Name To:

GRID

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
&AttractionId	&AttractionName		&trips	&update2	&newTrip

Total Trips:

Country Name:

Attraction Name From:

Attraction Name To:

GRID

Attraction Id	Attraction Name	Attraction Photo	Trips	&update2	&newTrip
AttractionId	AttractionName		&trips	&update2	&newTrip

Total Trips:

En el web panel sin tabla base, sencillísimo: condicionáramos el comando Load de acuerdo al valor de la variable &trips. Solo cargamos una línea en el grid si la variable &trips es mayor que 1. En caso contrario no la cargamos.

Pero, ¿y en el web panel con tabla base? Parecería que aquí no podríamos hacerlo, dado que el Load se ejecuta cuando se está por cargar la línea del grid, y parecería que no tenemos manera de detener la carga, porque el comando Load que en el otro explicitamos, aquí es implícito; lo coloca GeneXus en su código fuente, pero no nosotros. Sin embargo, sí podemos hacerlo. Si programamos el Load explícitamente, entonces GeneXus entiende que no deberá colocar el suyo. Lo deja en nuestras manos, digamos.

Probemos.

GeneXus[™]

training.genexus.com
wiki.genexus.com