

Telas web

Tabela base e navegação em Web panel com UM grid

GeneXus™

Web Panel com um Grid

Passemos agora ao caso de web panel com um grid.

The screenshot displays a web application interface for viewing country information. 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 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", "Trips", and "&update", and a "Total Trips" field labeled "&totalTrips".

Live Application View (Right): Titled "Application Name", it shows the rendered interface. The "Country Name" is set to "FRANCE". Below the search fields, a table displays attractions for France:

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

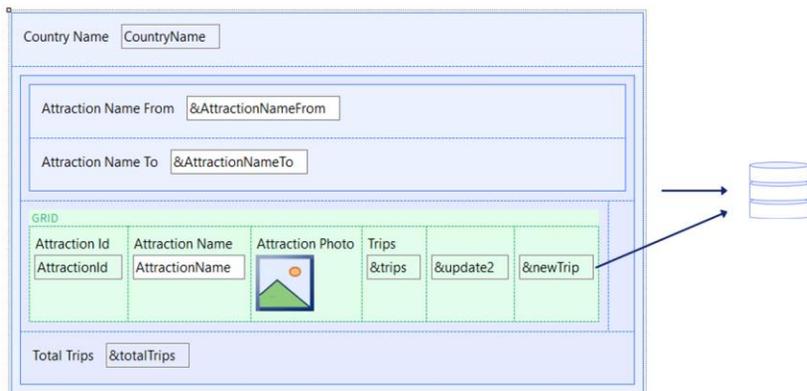
Below the table, the "Total Trips" is displayed as 0.

Aqui estamos recebendo por parâmetro o identificador de um país e mostrando em um grid todas as suas atrações turísticas, e na parte fixa, o nome desse país.

With one Grid

Web Form **Rules** Events Conditions Variables

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

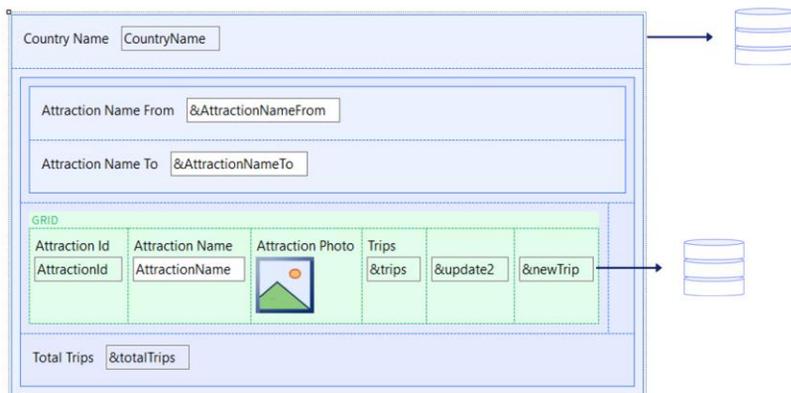


Quando o web panel tem no máximo um grid, podemos dizer que se existir tabela base, não há diferença entre falar da tabela base do web panel e da tabela base do grid. Por quê?

With one Grid

Web Form **Rules** Events Conditions Variables

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



Porque não poderá acontecer, como ocorre no caso dos Panels, ou seja, os desenvolvidos para executar com Angular, por exemplo, ou com Android ou Apple, que exista uma navegação para a parte fixa do panel e outra diferente para o grid.

With one Grid

Web Form **Rules** Events Conditions Variables

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

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

Total Trips &totalTrips

Para os web panels com um grid, se houver tabela base, será única, e o que deverá acontecer se estiver bem implementado, é que os atributos da parte fixa deverão ter o mesmo valor para todos os registros da tabela base que vão ser carregadas no grid, porque caso contrário, deveriam estar no grid e não na parte fixa!

With one Grid

Web Form **Rules** Events Conditions Variables`1=param(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	
Save State	False
Data Selector	(none)

Grid1's Conditions

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

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

Aqui serão carregadas no grid todas as atrações que passam pelos filtros das conditions e de pertencer ao país recebido por parâmetro. Portanto, para todas elas, o CountryName será o mesmo e por isso está na parte fixa.

Observemos que estaremos recebendo por parâmetro no atributo CountryId que NÃO PARTICIPA na determinação da possível tabela base, mas sim atua como filtro sobre as navegações que GeneXus encontra, tal como vimos no 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)

extended table

Grid1's Conditions	
AttractionName >=	&AttractionNameFrom when not &AttractionNameFrom.IsEmpty();
AttractionName <=	&AttractionNameTo when not &AttractionNameTo.IsEmpty();

Quais partes do web panel são levadas em conta para determinar tabela base? Evidentemente, ao aparecer um grid, além dos lugares que já havíamos visto para o caso anterior, soma-se tudo relativo ao grid.

Portanto, se ali aparecerem atributos, ou nas propriedades Order, conditions, Unique, Data Selector e, nem falar, se aparece especificada transação base na propriedade correspondente, então o grid (e, portanto, o Web panel) terá uma tabela base implícita.

Naturalmente, se existir transação base, deverá cumprir que todos os atributos do form e os que estejam em eventos "soltos", deverão pertencer à sua tabela estendida. Caso contrário, GeneXus nos advertirá quais atributos não poderão ser "alcançáveis" e ficarão sem valor.

Se não foi especificada transação base, então, como no caso do web panel sem grid, GeneXus a encontrará procurando a mínima tabela estendida que contenha todos os atributos que apareçam nos lugares que definimos. Sua tabela base será a tabela base escolhida.

With one Grid

Web Form **Rules** Events Conditions Variables

1 parm(in: CountryId);

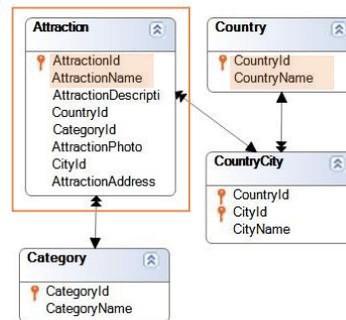
Country Name

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	Grid1's Conditions
Save State	False
Data Selector	(none)

Grid1's Conditions

```

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

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

Antes de passar a observar os eventos, por já existir transação base, podemos dizer que a tabela base será Attraction. E confirmamos que todos estes outros atributos pertencem à sua tabela estendida. Então, se não houvesse nem um único evento programado, saberíamos que será percorrida a tabela Attraction, ordenando-a por CountryId e AttractionName, filtrando pelas atrações cujo AttractionName satisfaça estas duas condições, e também cujo país corresponda ao recebido por parâmetro.

Digamos uma última vez: para os web panels, se há grid, os atributos da parte fixa correspondem à navegação do grid. Depois veremos que se há vários Grids, corresponderão à navegação de um deles.

Se não estivessemos recebendo no atributo CountryId, então, dado que serão carregadas atrações de diferentes países no grid, que valor assumirá o atributo CountryName na tela? É fácil suspeitar que será o primeiro ou o último dos registros carregados no grid.

Testemos.

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

Aqui temos o web panel corretamente programado. Vejamos, antes, sua lista de navegação.

Vemos que, efetivamente, há uma tabela base, que é Attraction. A vemos claramente identificada no nível do evento Load do grid.

E agora vamos executar antes de modificar o parâmetro. A partir deste outro web panel invocamos o nosso.

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		
Louvre Museum		0	Update	New trip
The Great Wall		0	Update	New trip
Eifel Tower		0	Update	New trip
Clint the Redemmer		0	Update	New trip
Smithsonian Institute		0	Update	New trip
Matisse Museum		0	Update	New trip
Forbidden city		0	Update	New trip
Goldenman Viaduct		0	Update	New trip
Meet the Emperor		0	Update	New trip

Agora modifiquemos nosso painel para que em vez de receber no atributo CountryId, se receba na variável &CountryId. E, deste modo, não haja um filtro automático.

Observemos a lista de navegação. Vemos que agora não haverá filtro.

E vamos vê-lo em execução. Vemos que carregou como nome de país o do primeiro registro, não o do 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

```

Bem, deixemos o parâmetro como o tínhamos.

Se agora vamos observar os eventos, reparamos no fato de que, evidentemente, o evento Load se programou já sabendo que o grid teria tabela base. Se analisarmos como faz GeneXus, quais atributos participarão na determinação da tabela base, vemos que são todos estes que estão "soltos".

Claramente TripDate não. Este só participa na determinação da tabela a ser navegada pela fórmula count. Evidentemente, assim como ocorre com o web panel em geral, se o grid tem tabela base então esta fórmula count também poderá ter uma condição implícita, que, neste caso, é que contará as trips da atração turística que está para ser carregada.

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
  &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, não basta substituir todos os atributos que mostramos por variáveis em todos os lados se esquecemos de fazê-lo em um destes lugares. Por exemplo, aqui fizemos um Save as do panel anterior para implementá-lo sem tabela base.

Trocamos todos os atributos do form por variáveis, e também esses mesmos atributos nos eventos.

De fato, programamos o evento Load do grid especificando o For each para percorrer a tabela Attraction e carregar todas as variáveis do grid (e também a da parte fixa, &CountryName), pensando que estamos programando o web panel sem tabela base.

Não especificamos a transação base para o For each, mas poderíamos tê-lo feito. Sabemos que será filtrado por CountryId, porque está recebendo por parâmetro e, além disso, que esse atributo da regra parm não participa em determinar tabela base.

Mas, onde estamos especificando os filtros por nome de atração? Os esquecemos. Nos esquecemos de modificar também as propriedades do grid! E as propriedades do grid determinam tabela 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)

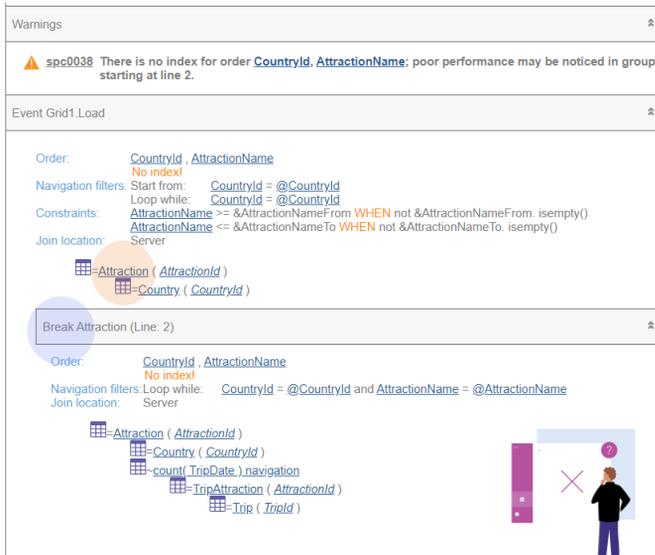
Break Attraction (Line: 2)

Order: CountryId, AttractionName
No index

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

Join location: Server

Attraction (AttractionId)
Country (CountryId)
country(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 fato, o que nos diz a lista de navegação. Justamente, encontrou tabela base Attraction, então executará o Load N vezes, filtrando pelo país recebido por parâmetro e também pelas variáveis de acordo com as conditions que especificamos no grid. E para cada registro desses, executará o For each do Load, realizando um corte de controle, porque a tabela base é a mesma, Attraction.

Tudo por esquecer de revisar as propriedades do 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

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
 Constrains: 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)

Então o que devemos fazer é levar estas propriedades ao For each.

Primeiro a transação base...

Depois o order...

E por último as conditions, como cláusulas where...

Gravamos. Está nos dando este error porque não tem ponto e vírgula na sintaxe do for each. Voltamos a gravar. Agora sim.

E vejamos agora a lista de navegação.

Agora sim, conseguimos que o web panel não tivesse tabela base.

Vejamos que se colocássemos as cláusulas where como Conditions, mas não do grid, e sim como conditions gerais do Web panel, como acontecia no caso de um panel sem grid, os atributos destas condições não participam em determinar tabela base. Aqui vemos claramente. A lista de navegação não difere da anterior.

O problema das conditions gerais é que vão aplicar-se como filtros em toda a navegação do web panel que envolve seus atributos e, às vezes, isso não nos interessa. Pensemos, por exemplo, no caso de ter múltiplos 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

```

Country Name

Attraction Name From

Attraction Name To

GRID

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

Only load
an
attraction
if &trips > 1
?

```

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

```

Country Name

Attraction Name From

Attraction Name To

GRID

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

Em geral, é melhor prática implementar os web panels com tabela base, já que nos poupa de todo este trabalho. Aqui tivemos que estabelecer o for each, e atribuir-lhe às variáveis os valores dos atributos e especificar o comando Load para carregar cada linha.

Em vez disso, na versão com tabela base, vemos que tudo foi muito mais simples.

Mas, como faríamos se quiséssemos apenas carregar atrações que estejam em mais de uma excursão (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

```

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:

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:

No web panel sem tabela base, muito simples: condicionaríamos o comando Load de acordo com o valor da variável &trips. Só carregamos uma linha no grid se a variável &trips é maior que 1. Caso contrário, não a carregamos.

Mas, e no web panel com tabela base? Pareceria que aqui não poderíamos fazê-lo, dado que o Load se executa quando se está prestes a carregar a linha do grid, e pareceria que não temos como deter a carga, porque o comando Load que, no outro, explicitamos, aqui é implícito; GeneXus o coloca em seu código fonte, mas nós não. No entanto, podemos fazê-lo. Se programamos explicitamente o Load, então GeneXus entende que não deverá colocar o seu. Deixa-o em nossas mãos, digamos.

Testemos.

GeneXus[™]

training.genexus.com
wiki.genexus.com