

A web panel é o objeto mais flexível que GeneXus oferece.

Como já temos visto em alguns exemplos que temos mostrado, toda web panel oferece um web form, que é uma página web que nos permite "desenhar" e disponibilizar várias funcionalidades.

EnterAttractionsFilter ×	-
Web Form Rules Events Conditions Variables	
✓ <no action="" group="" selected=""></no>	
MainTable	→
Country Id &CountryId Attraction Name From &AttractionNameFrom Attraction Name To &AttractionNameTo List Attractions By Country List Attractions By Name	

Nesse exemplo havíamos visto que o fato de incluir variáveis no web form tornam as variáveis habilitadas para que o usuário entre com algum valor. Eram controles de entrada, ou também conhecidos como não readonly.

Essa variável em particular, do tipo combo dinâmico

EnterAttractionsFilter ×	O Properties	4 ×
Web Form Rules Events Conditions Variables	General Class	
No action group selected>	🔠 🦺 🌾 🛛 Filter	×
	✓ Attribute/Variable: &Cor	untryld
	Control Name	&Countryld
QQQ	Attribute	&Countryld
Country Id &CountryId ~	Label Position	Left
	Label Caption	Country Id
	Readonly	False
Attraction Name From &AttractionNameFrom	Return On Click	False
	> Appearance	
	✓ Control Info	
Attraction Name To &AttractionNameTo	Control Type	Dynamic Combo Box
	Data Source From	Attributes
	Item Values	Countryld
List Attractions By Country List Attractions By Name	Item Descriptions	CountryName
	Sort Descriptions	True
	Conditions	
	Instantiated Attributes	
	Empty Item	False
	Notify Context Change	False

esperava que o usuário escolhesse um país dos carregados no combo e ao pressionar o botão "List Attractions By Country", executava-se o evento associado...

EnterAttractionsFilter * ×	
Web Form Rules Events* Conditions Variables	
Events 🗸	
1 Event 'List Attractions By Country' 2 AttractionsList(&CountryId) 3 //AttractionsReport(&CountryId) 4 -Endevent	- 1 <mark>1111</mark>
5 6 ⊡ Event 'List Attractions By Name' 7 AttractionsByName(&AttractionNameFrom, &AttractionNameTo) 8 - Endevent 9	
	>

...chamando o pdf que listava as atrações desse país.

E nessas outras variáveis o usuário entrava com um range de nomes de atração para que, pressionando esse outro botão,

EnterAttractionsFilter * ×	*
	
MainTable ListAttractionsByName	<i>></i>
Country Id &CountryId ~	
Attraction Name From &AttractionNameFrom	
Attraction Name To &AttractionNameTo	
	,

fosse chamado o relatório pdf que mostrava as atrações dentro desse range recebido via parâmetro.



Lembremos o layout:

AttractionsByName ×	-
Source Layout Rules Conditions Variables	
Image: A state of the state	1 1 1 1 1 1 1 7 1 1
🗆 🚍 Title	
•	
Attractions List	
Auacuons List	
E 🖂 🚍 Column Titles	
id Name Country Photo	
Attraction	Photo
Attraction Attraction Name Country Name	Photo
Attraction AttractionName CountryName	Photo
Attraction AttractionName CountryName	iPhois
Attraction AttractionName CountryName	iPhote
Attraction AttractionName CountryName	iPhote

E no Source programávamos a consulta a base de dados com o for each, filtrando por nome:

+ AttractionsByName ×	-
Source Layout Rules Conditions Variables	
Subroutines 🗸	
1 print Title	15
2 print ColumnTitles	
3 ⊡ For each Attraction order CountryName	_
4 where AttractionName >= &NameFrom	
5 where AttractionName <= &NameTo	
6 print Attractions	
7 - endfor	
8	
9	
10	
11	
	>

Mas por que definir essas consultas através de relatórios pdf e não diretamente na própria tela na qual solicitamos ao usuário os dados para os filtros?

Por que não colocar aqui, no lugar dos botões, um grid que mostre as atrações desejadas?

EnterAttractionsFilter ×
Web Form Rules Events Conditions Variables
✓ <no action="" group="" selected=""></no>
✓ Image: MainTable
Country Id &CountryId ~
Attraction Name From & AttractionNameFrom
Attraction Name To &AttractionNameTo
List Attractions By Country List Attractions By Name

As web panels, além de permitir definir variáveis para utilizar nas ações programadas nos botões, permitem, -e é seu objetivo fundamental- implementar consultas **interativas** na base de dados.

A terminologia **"interativas"** refere-se a que o usuário pode informar na página da web panel uma e outra vez diferentes valores **–nas variáveis-** e consultar na sequência dados da base de dados que correspondam com esses valores informados, utilizando-os como filtros, como veremos agora.

Salvemos essa web panel com outro nome. Como o que vamos implementar vai ser similar a um work with, chamaremos:

New Object		×
Select a Category:	Select a Type:	
Common Workflow Reporting Documentation Web Smart Devices	Reb Panel	
WILL DO TO LA		
Web application form. Inte	eract with end user using events, show or request data and more.	
Name:	WWAttractionsFromScratch	
Description:	WWAttractions From Scratch	
	Create Ca	incel

Eliminemos os botões, que já não serão necessários, assim como os eventos associados. Agora inserimos debaixo das variáveis um controle do tipo grid:



Abre-se essa tela para escolher os atributos e/ou variáveis que serão as colunas desse grid.

Insert	Attribute/Variable				×
Fil	ters				
Pati Obj Obj	tem:			 ▼ ▼ 	
	Name	Module	Туре	Description	^
&	Countryld		Attribute:Country	Country Id	
&	AttractionNameFro		Attribute:Attractio	Attraction Name	
&	AttractionNameTo		Attribute:Attractio	Attraction Name	
&	Today		Date	Today	
&	Time		Character(8)	Time	
&	Pgmname		Character(128)	Pgmname	
&	Pgmdesc		Character(256)	Pgmdesc	1
\$	AirlineDiscountPer		Percentaje	Airline Discount	
\$	Airlineld		ld	Airline Id	1
\$	AirlineName		Name	Airline Name	
\$	AirportId		ld	Airport Id	1
\$	AirportName		Name	Airport Name	1
\$	AttractionAddress		Address, GeneX	Attraction Address	1
\$	AttractionId		ld	Attraction Id	1
\$	AttractionName		Name	Attraction Name]
Ø	AttractionPhoto		Image	Attraction Photo	1
\$	CategoryId		ld	Category Id	
Ô	CategoryName		Name	Category Name	4
Ne	w Variable Ed	Jit		OK Cancel	

Como o que queremos é o mesmo que mostrávamos no relatório pdf...

♣ AttractionsByName ×	•
Source Layout Rules Conditions Variables	
D • • • 1 • • • 1 • • • 1 • • • 2 • • • 1 • • • 3 • • • 1 • • • 4 • • • 1 • • • 5 • • • 1 • • • 6 • • • 1 • • • 7 •	
🖸 🗇 🚍 Title	
Attractions List	• • • • • • • • • • • • • •
E Column Titles	
id Name Country Photo	
E 🚍 Attractions	
- AttractionPhoto	••••••
Attra AttractionName CountryName	• • • • • • • • • • • • • •
-	
	>
	Zoom Factor: 100

...escolhemos os atributos AttractionId, AttractionName, AttractionPhoto e CountryName...

Inse	ert Attribute/Varial	ble					×
F	ilters						
_		[7	
P	attem:						
0	bject Type:					r	
0	bject:					r	
	Name	Modu	ıle	Туре	Desc	ription	^
8	Pgmdesc			Character(256	5) Pgm	lesc	
8	AirlineDiscount	tPer		Percentaje	Airlin	e Discount	
8	Airlineld			ld	Airlin	ie Id	
8	AirlineName			Name	Airlin	e Name	
8	AirportId			ld	Airpo	vrt Id	
8	AirportName			Name	Airpo	ort Name	
8	AttractionAddre	ss		Address, Gen	ieX Attra	ction Address	
8	AttractionId			ld	Attra	ction Id	
×.	AttractionName	•		Name	Attra	ction Name	
Ę	AttractionPhoto			Image	Attra	ction Photo	
×.	CategoryId			ld	Cate	gory Id	
8	CategoryName			Name	Cate	gory Name	
8	3 Cityld			ld	City	d	
8	CityName			Name	City	Name	
8	CountryId			ld	Cour	itry Id	
8	CountryName			Name	Cour	try Name	
8	CustomerAdde	dDa		Date	Cust	omer Added	
8	CustomerAddre	ess		Address, Gen	ieX Cust	omer Address	×
N	lew Variable	Edit			ОК	Cance	1

e pressionamos OK. Visualizamos que foi criado um grid com essas colunas. Movamos CountryName para aqui:

EnterAttractionsFilter * × 🔜 WWAttractionsFromScratch * ×	•				
Web Form * Rules Events * Conditions Variables					
MainTable EGrid1 CountryName	÷				
Country Id &CountryId ~					
Attraction Name From &AttractionNameFrom					
Attraction Name To &AttractionNameTo					
GRID					
Attraction Id Attraction Name Country Name Attraction Photo					
AttractionId AttractionName CountryName					

Podemos alterar os títulos de cada coluna, editando as propriedades de cada um dos atributos que pertencem as colunas do grid.

EnterAttractionsFilter * X 🔜 WWAttractionsFromScratch * X	-	O Properties	7 ×				
Web Form * Rules Events * Conditions Variables	General Class						
✓ <no action="" group="" selected=""></no>		🔠 🛃 🌾 Filter	×				
		Attribute/Variable: A	AttractionPhoto				
	6	Attribute	AttractionPhoto				
	\rightarrow	Title	Photo				
Country Id &CountryId V		Class	ImageAttribute				
		Column Class					
Attraction Name From &AttractionNameFrom		Return On Click	False				
		On Click Event					
Attraction Name To RiAttractionNameTo		✓ Control Info					
		Control Type	Image				
		Notify Context Cha	False				
GRID		Maximum Upload	Large				
Id Attraction Name Country Photo		> Behavior					
		✓ Appearance					
		Auto Resize	True				
		Visible	True				
	_	Alternate Text					
🛨 Output 🔫	×	Tooltip Text					

Pressionemos F5 para executar nossa nova web panel assim como temos feito até agora. Os **atributos** no form de uma web panel são, por padrão, **de saída**. São **readonly**. Isso significa que GeneXus interpreta que deve ir a base de dados buscar seu valor para mostrar ao usuário.

$${}^{\rm Page}8$$

WWAttractions From Scra	at x						*	-		×
🗲 🔶 C 🗋 trialapp	os3.genexus.c	com/Id85	62acf4c97c	0fe8095a5	ic284d688	e07/wwa]☆	2	<u>a</u> 0	Ξ
Applicat	tion Na	ame					by	Ge	neXu:	Ŝ
Recents W	WAttractions	From								
Country Id	Brazi	I	•							
Attraction Name From										
Attraction Name To										
Id Attraction Name	Country	Photo								
22 Louvre Museum	France									
23 The Great Wall	China	XX								
24 Eiffel Tower	France	Å								
25 Christ the Redemme	r Brazil	TT.								
26 Smithsonian Institute	United States	s Antitio n								
27 Matisse Museum	France									
28 Forbidden city	China									

Visualizamos que saíram impressas todas as atrações, com os dados que indicamos (o id, nome, país e foto). Além do que, saíram ordenadas por AttractionId.

Tão somente em colocar um grid com esses atributos, GeneXus entendeu que deveria ir a base de dados, acessar a tabela Attraction, acessar também a tabela Country para trazer o país da atração, tal como fazíamos com o comando for each (sem essas cláusulas):

🔜 WWAttractionsFromScratch X 📋 Navigation View X 🔹 🔹	•) AttractionsByName ×
Web Form Rules Events Conditions Variables	Source Layout Rules Conditions Variables
	Subroutines 🗸
MainTable	1 print Title 2 print ColumnTitles
Country Id &CountryId V	4 5 minute attraction and a Standborg and 6 print Attractions
Attraction Name From &AttractionNameFrom Attraction Name To &AttractionNameTo	7 endfor 9 10 11 =
GRID C	
Id Attraction Name Country Photo AttractionId AttractionName CountryName	
< >	< > >



Se observamos as propriedades do grid, aparece de fato uma de nome **Base Trn**, que é análoga a transação base do for each. De fato, para assegurarmos de que para o grid seja eleita a tabela base Attraction, que é a que desejamos, é uma boa prática indicar a transação base, igual ao que é feito para o for each.

O Properties	7 ×
General Class	
🔠 灯 🌾 🛛 Filter	×
✓ Grid: Grid1	
Control Name	Grid1
Collection	
Base Trn	Attraction
Order	
Conditions	
Data Selector	(none)
✓ Appearance	
Class	Grid
Custom Render	
Empty Grid Text	
Auto Resize	True
Width	
Height	
Rows	0
Tooltip Text	
> Layout	
> Behavior	

Por outro lado, observemos que temos uma propriedade **Order** para o grid. Essa propriedade corresponde-se com a cláusula Order do for each.

Assim, se quisermos ordenar por nome de país, como no relatório:

🔜 WWAttractionsFromScratch * 🗙 🛅 Navigation View X 🔹 • •	AttractionsByName ×	O Properties
Web Form * Rules Events Conditions Variables	Source Layout Rules Conditions Variables	General Class
+ <no action="" group="" selected=""></no>	Subroutines 🗸	Sill 21 Ye Filter
MainTable	1 print Title 2 print ColumnTitles 3 Ecc. acc Attraction order CountryName	Control Name Grid1
Country ld &Countryld ~	4 5 where Attraction on the Contribution	Collection Base Trn Attraction
	6 print Attractions 7 - endfor	Order
Attraction Name From &AttractionNameFrom	9	Data Selector (none)
Attraction Name To &AttractionNameTo	Gnd1's Order	
o Attraction Name Country Photo CountryName CountryName CountryName O		OK Cancel
<	×	Rows 0 Tooltip Text

 $_{\rm Page}10$

Pressionemos F5.

WWAttractions From Scr	at X							^	-		×
← → C 🗋 trialap	ps3.genexus.co	m /ld85	62acf4c9	7c0fe80	095a5c2	284d688e	e07/wwatti	rac 🔄 🏠	ا ا	e 0	Ξ
Applica	tion Na	me						by	Ger	neXu	Ŝ
Recents V	VWAttractions	From									
Country Id	Brazi	I	•								
Attraction Name From											
Attraction Name To											
Id Attraction Name	Country	Photo									
25 Christ the Redemme	Brazil	Tr.									
23 The Great Wall	China	Nor									
28 Forbidden city	China										
22 Louvre Museum	France	-									
27 Matisse Museum	France										
24 Eiffel Tower	France	Å									
26 Smithsonian Institute	United States	a a a a a a a a a a a a a a a a a a a									

E visualizamos que agora o grid sai ordenado por nome de país.

Observemos o relatório de navegação da web panel:



Aqui está indicando a navegação que tem que realizar para carregar o combo box da variável &Countryld, que por agora não estamos usando para nada.

E aqui está indicando a navegação que terá que realizar para carregar (Load) o grid. Vemos que o que lista para essa carga é idêntico ao que lista para um for each. Vemos que escolheu a tabela Attraction, lendo-a por CountryName, o atributo da propriedade Order. Lerá toda a tabela. E para cada registro de Attraction a ser carregado, acessará a tabela Country para mostrar o CountryName da atração.

Até agora não fizemos nada com as variáveis. Mas queríamos utlizá-las para filtrar os dados que são mostrados no grid, tanto por país como por nome de atração.

Em AttractoinsList filtrávamos por país. Como indicamos esse filtro para o grid? Através da propriedade **Conditions**:



Aqui estamos dizendo ao GeneXus que queremos que quando leia a tabela base do grid, Attraction, filtre os registros para os quais o Countryld da atração é igual ao valor que o usuário tem escolhido no combo &Countryld.

Pressionemos F5.

WWAttractions From Scrat			1	-		×
← → C 🗋 trialapps3.ge	nexus.com/ld8	562acf4c97c0fe8095a5c284d688e07/ww	: \$	2	<u>e</u> 0	Ξ
Application	Gei	neXu	Ŝ			
Recents WWAttra	ctions From					
Country Id	Brazil	Y				
Attraction Name From						
Attraction Name To						
Id Attraction Name Coun 25 Christ the Redemmer Brazil	try Photo					

Page J

Observemos que por padrão, o combo tem o valor Brasil, e que no grid só visualizamos atrações do Brasil.

Se escolhemos França, vemos que a tela é atualizada, voltando-se a carregar o grid, agora com as atrações da França.

WWAttractions From Scra X	4	-		×
← → C [] trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/w	/W2 🔄 🖒	2	<u>a</u> (≡
Application Name	by	Gei	neXu	Ŝ
Recents WWAttractions From				
Country Id France •				
Attraction Name From				
Attraction Name To				
Id Attraction Name Country Photo				
22 Louvre Museum France				
24 Eiffel Tower France				
27 Matisse Museum France				

É bem provável que queremos que o combo apareça sem valor escolhido na primeira vez, e que nesse caso seja mostrado as atrações de todos os países.

Para isso, editemos as propriedades do combo box... e alteremos para True a propriedade de nome Empty Item. Isso fará que seja adicionado uma opção "(none)" ao combo. Corresponderá ao valor empty, ou seja, vazio, zero.



Então, vamos até a propriedade **Conditions** e especifiquemos que queremos que seja aplicada essa condição somente quando o combo não tem o valor vazio. Quando tenha valor vazio, que não aplique-a.

EnterAttractionsFilter × 🔜 WWAttractionsFromScratch * × 🗐 Navi <	• •	Attractions	ByName ×	🕻 Attractionsl < 🕨 🔻	0	Properties	7 ×
Web Form * Rules Events Conditions Variables		Source Lay	out Rules	Conditions Variables	G	eneral Class	
		Subroutines	~	-	>	Al 🏹 Filter	×
		1 1	rint Title			Grid: Grid1	
	Ľ	2	rint Colum	nnTitles 📲		Control Name	Grid1
		3 🗆 F 4	or each At	cuntryId = &Count		Collection	
Country Id &CountryId ~		5	print A	Attractions		Base Trn	Attraction
		6 L e	ndfor			Order	CountryName
Attraction Name From &AttractionNameFrom		8				Conditions	Countryld = &Countryld v
Attraction Name To &AttractionNameTo		9 10 1	irid1's Condit Country: <	ions I <u>d</u> = &CountryId wher	n not	t &CountryId.IsEmpty()	× ;
AttractionId AttractionName CountryName		L			Ľ	Width	
				_		Rows	0

Executemos...

Aparece o valor (None) no combo, e além disso para esse caso, as atrações não estão sendo filtradas. Todas são mostradas:

WWAttractions From Scr	rat X					1	-		×
🗲 🔶 C 🗋 trialap	ps3.genexus.	com/Id8562	acf4c97c0fe?	8095a5c2	34d688e07/w	w: 😰 🏠	<mark>ا ا</mark>	a 0	Ξ
Applica	tion Na	ame					Ger	neXu.	Ŝ
Recents W	/WAttractions	From							
Country Id	(Non	e) 🔹							
Attraction Name From									
Attraction Name To									
Id Attraction Name	Country	Photo							
25 Christ the Redemme	er Brazil	11							
23 The Great Wall	China	X							
28 Forbidden city	China								
22 Louvre Museum	France								
27 Matisse Museum	France								
24 Eiffel Tower	France	A							
26 Smithsonian Institute	e United State	s militin							

Se agora escolhemos, por exemplo, França:

$$_{\rm Page}14$$

WWAttractions From Screit ×	2	-		×
← → C C Trialapps3.genexus.com/Id8562acf4c97c0fe8095a5c284d688e07/www	12	2	A 0	Ξ
Application Name		Gei	neXu:	Ŝ
Recents WWAttractions From				
Country Id France				
Attraction Name From				
Attraction Name To				
Id Attraction Name Country Photo				
22 Louvre Museum France				
24 Eiffel Tower France				
27 Matisse Museum France				

Como a variável não tem valor vazio, o filtro é aplicado e são mostradas as atrações da França.

Também teríamos que adicionar os filtros por nome da atração, que gostaríamos que fosse somado ao outro filtro.

Então... se no relatório filtrávamos no for each com essas duas cláusulas where... no grid adicionaremos como condições:



AttractionName maior ou igual ao valor da variável &AttractionNameFrom do form, que o usuário haverá digitado, se desejar. Outra vez, se o usuário não entra com valor na variável, não queremos que esse filtro seja aplicado. Então usamos a cláusula when. Essa cláusula também pode ser utilizada na cláusula where do for each, de uma forma completamente análoga.

Page 1

Gri	id1's Conditions	×
	<pre>CountryId = &CountryId when not &CountryId.IsEmpty();</pre>	
	<pre>AttractionName >= &AttractionNameFrom when not &AttractionNameFrom.IsEmpty();</pre>	<
		Y
1<	>	
	OK Cancel	

E adicionamos o outro filtro:

Gr	id1's Conditions	×
	CountryId = &CountryId when not &CountryId.IsEmpty();	
	AttractionName >= &AttractionNameFrom when not &AttractionNameFrom.IsEmpty();	^
	<pre>AttractionName <= &AttractionNameTo when not &AttractionNameTo.IsEmpty();</pre>	~
<	>	
	OK Cancel	

Executemos novamente. E escolhemos ver as atrações entre A e F:

WWAttractions From Scratt ×	WWAttractions From Scrat	▲ – □ ×
← → C 🗋 trialapps3.ge	nexus.com/ld8562acf4c97c0fe8095a5c284d688e07/w	vwa 🗟 🖧 🔊 🖶 🚺 🗏
Applicatior	n Name	by GeneXuš
Recents WWAttra	ictions From	
Country Id	(None)	
Attraction Name From	А	
Attraction Name To	F	
Id Attraction Name Cou	ntry Photo	
25 Christ the Redemmer Braz		
24 Eiffel Tower Fran	ce	

Podemos solicitar na web panel que se o usuário escolhe um país, então ordene a informação por Countryld e dentro de Countryld por AttractionName, caso contrário, ordene por AttractionName. Isso pensando em otimizar a busca dos registros da tabela.

Para isso, editamos no grid a propriedade Order e escrevemos primeiro a ordem, condicionada a que o usuário tenha escolhido um país no combo. Nesse caso será filtrado por esse país, mas além disso, as atrações sairão listadas alfabeticamente para esse país. E se o usuário deixar o combo com o valor "(none)", isto é, vazio, então será escolhida a ordem seguinte, que é por AttractionName:

Gr	id1's Order		×
	CountryId, AttracionName when not &CountryId.IsEmpty() AttractionName		
			^
<		>	~
	OK Cancel]

Não nos deteremos nisso aqui. Deixamos apontado unicamente para mostrar que também é possível condicionar a forma em que se quer ordenar a informação. Isso é idêntico a um for each.

Executemos:

WWAttractions From Scra	at x							*	-		×
🗲 🔶 C 🗋 trialapp	os3.genexus.c	om/Id856	2acf4c97	7c0fe80	95a5c2	84d688e0)7/wwa	s 2	2	a 0	Ξ
Applicat	tion Na	me						by	Gen	neXu.	Ŝ
Recents W	WAttractions F	From									
Country Id	(None	e) •	•								
Attraction Name From											
Attraction Name To											
Id Attraction Name	Country	Photo									
25 Christ the Redemme	r Brazil	T.									
24 Eiffel Tower	France	Å									
28 Forbidden city	China										
22 Louvre Museum	France										
27 Matisse Museum	France										
26 Smithsonian Institute	United States	in in									
23 The Great Wall	China										
					_						

Aqui ordenou-se por AttractionName. E se escolhermos França, será ordenado pelo Id da França e dentro dele por AttractionName.

WWAttractions From Scrall X	*	-		×
← → C [] trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/ww&	3	2	& 0	Ξ
Application Name		Gei	neXu	Ŝ
Recents WWAttractions From				
Country Id France •				
Attraction Name From				
Attraction Name To				
Id Attraction Name Country Photo				
24 Eiffel Tower France				
22 Louvre Museum France				
27 Matisse Museum France				

Em definitivo, sempre vemos as atrações ordenadas alfabeticamente.

Se dentro das atrações da França queremos as que se encontram entre A e F:

WWAttractions From Scral									*	-			×
← → C 🗋 trialapps3.ge	nexus.com/ld8	8562act	cf4c9	7c0fe	e8095a	a5c284	d688e	e07/ww	/₂◙∑	2		0	Ξ
Applicatior	n Name	ļ							b	, Ge	ne)	(US	5
Recents WWAttra	ctions From												
Country Id	France	٣											
Attraction Name From	А												
Attraction Name To	F												
Id Attraction Name Country	Photo												
24 Eiffel Tower France	Å												

Vemos que o grid foi carregado filtrando pelas três condições que havíamos escrito:



Resumindo o que foi feito até o momento:

Temos implementado uma web panel na qual incluímos algumas variáveis para que o usuário entre com valor, e inserimos um controle Grid com atributos.

Os atributos correspondem a informação da base de dados, pelo que GeneXus entende que deve ir buscar essa informação. Um grid com atributos é como um for each. Por isso contamos com a propriedade **Base Trn**, como a do for each, para especificar o nível da transação cuja tabela associada queremos ler. Chamamos essa tabela, **tabela base do grid**. Caso não especifiquemos, como também ocorre com um for each, GeneXus infere na base os atributos que são utilizados. Esse caso não veremos aqui.

Todos os atributos do grid deverão pertencer, como no caso de um for each, à tabela estendida dessa tabela base.

Assim como em um for each ordenamos a informação com a cláusula Order e filtramos os dados a serem devolvidos pela consulta com uma ou várias cláusulas where, para fazer o mesmo com o grid temos as propriedades Order e Conditions, respectivamente.

Agora bem, em um for each, programamos o que queremos que seja feito com cada registro que cumpra as condições, dentro do seu corpo.

	GeneXus
<no action="" group="" selected=""></no>	
Country Id &CountryId > endfor	
Attraction Name From &AttractionNameFrom	
Attraction Name To &AttractionNameTo	
GRID •	
Id Attraction Name Country Photo AttractionId AttractionName CountryName Image: CountryName	

Por exemplo, no relatório de atrações turísticas, o comando print Attraction imprimia na saída o que em nosso caso seria a linha do grid. No caso do grid não é necessário especificar. Isso é feito automaticamente.

	GeneXus
 <no action="" group="" selected=""></no> MainTable Grid1 Country Id &CountryId ~ 	For each <u>Main_Code</u> Print Attractions endfor
Attraction Name From &AttractionNameFrom Attraction Name To &AttractionNameTo	Atractore Atractor/Note Atractor/Note Atractor/Note
Attraction Name Country Photo	

Mas, por exemplo, imaginemos que temos uma transação Trip que registra as excurções oferecida pela agência de viagens. De forma muito simplificada, imaginemos que cada excurção só registra a data em que ocorrerá e a descrição e na sequência são registradas as atrações turísticas que serão visitadas por essa excursão. Bem, então agora suponhamos que no relatório de atrações turísticas queremos ver também a quantidade de trips que tem associada cada atração. Para isso bastava definir uma variável &trips, numérica, e atribuir dentro do corpo do for each, (isso é, quando o for each está posicionado no registro de sua tabela base que está para processar) o resultado de contar os trips dessa atração. E colocar essa variável no print block.

Page Z C

	GeneXus
<no action="" group="" selected=""></no>	For each <u>Main_Code</u> &trips = Count(TripDate) Print Attractions endfor
Attraction Name From &AttractionNameFrom	C The Attraction Parts Country Name State Star Prob
Attraction Name IO Country Photo AttractionName Country Photo AttractionName CountryName 0	Trip TripDate TripDescription Attraction

Para fazer o mesmo na web panel, clicamos com o botão direito sobre o grid e Insert Attribute/Variable

Web Form Rules Events Conditions	Variables
✓ <no action="" group="" selected=""></no>	
🕢 🏢 MainTable 🛛 🗮 Grid 1	>
Country Id &CountryId ~	
Attraction Name From & Attraction	lameFrom
Attraction Name To & Attraction Name	2010
GRID Id Attracti	ribute/Variable
Attraction Attracti X Delete	Del
O Part Ro	w Before
⇒E Insert Ro	w After
Insert Ce	l Before
insert Ce	l After
Delete Ro	
	211

Depois New Variable:

Insert	Attribute/Variable				×
Filt	ters				
Patt	em:				
Obie	act Type:			-	
Obie	ect:			-	
				*	
	Name	Module	Туре	Description	^
&	Countryld		Attribute:Country	Country Id	
&	AttractionNameFro		Attribute:Attractio	Attraction Name	
&	AttractionNameTo		Attribute:Attractio	Attraction Name	
&	Today		Date	Today	
&	Time		Character(8)	Time	
&	Pgmname		Character(128)	Pgmname	
&	Pgmdesc		Character(256)	Pgmdesc	
Ø	AirlineDiscountPer		Percentaje	Airline Discount	
Ø	Airlineld		ld	Airline Id	
Ø	AirlineName		Name	Airline Name	
۵	AirportId		ld	Airport Id	
Ô	AirportName		Name	Airport Name	
Ô	AttractionAddress		Address, GeneX	Attraction Address	
Ô	AttractionId		ld	Attraction Id	
\$	AttractionName		Name	Attraction Name	
\$	AttractionPhoto		Image	Attraction Photo	
ø	Categoryld		ld	Category Id	
Ô	CategoryName		Name	Category Name	¥ .
Nev	w Variable Ed	dit		OK Cance	1

E definimos a variável trips:

S=- A1		
🚛 🛃 🌾 Filter		X
 Variable: &trips 		^
Name	trips	
Description	trips	
Column title	trips	
Class	Attribute	
	Help	2
✓ Type Definition	1	
Based on	(none)	
Data Type	Numeric	
Length	4	
Decimals	0	
Signed	False	
Collection	False	
Dimensions	Scalar	
Initial value		

Ao pressionar OK... podemos ver que foi inserida como coluna do grid. Movemos ela para que ocupe a posição que nos interessa dentro do grid.

$${}^{\rm Page}22$$

Country Id	&Countryld ~]		
Attractio	n Name From الا	kAttractionNameFrom	1	
Attractio	n Name To &At	ttractionNameTo		
GRID		O		
trips &trips	ld AttractionId	Attraction Name AttractionName	Country CountryName	Photo

E mudamos a propriedade **Title** para que o título da coluna saia em maiúscula.

GRID				0
ld	Attraction Name	Country	Photo	Trips
AttractionId	AttractionName	CountryName		&trips

Isso corresponde a ter inserido a variável no printblock. Mas onde informamos como é calculada?

	GeneXus
	For each <u>Main_Code</u> <u>&trips = Count(TripDate</u> Print Attractions endfor
Country Id &CountryId ~	C 🖷 Attractions
Attraction Name From &AttractionNameFrom	Aira Atractontare Countrylane
Attraction Name To &AttractionNameTo	
GRID Id Attraction Name Country Photo AttractionId AttractionName CountryName Ration	



No for each é dentro de seu corpo. E aqui?

Contamos com o evento **Load** do sistema. Ali dentro programaremos o que queremos que seja executado quando se está posicionado em um registro da tabela base do grid, imediatamente antes que a linha correspondente seja carregada no grid.

			Ge	eneXus	5
Country Id &CountryId ~	For ea	ch ain_Code -	&trips = C Print Attra	ount(Tri actions	pDate
	AttractionId	AttractionName	Countryld	Cityld	
Attraction Name From &AttractionNameFrom	1	Louvre Museum	2	1	
	2	The Great Wall	3	1	
Attraction Name To &AttractionNameTo	3	Eiffel Tower	2	1	
GRID Id Attraction Name AttractionName CountryName CountryName CountryName	new line!	Event Load <u>Code</u> endevent			

Em nosso caso, ali é onde atribuiríamos valor na variável & Trips:

			Ge	eneXus	5
Trip TripDate TripDescription AttractionId AttractionName	For ea	ch ain_Code -	&trips = 0 Print Attra	Count(Trij actions	oDate)
	AttractionId	AttractionName	Countryld	CityId	
Attraction Name From &AttractionNameFrom	1	Louvre Museum	2	1	
	2	The Great Wall	3	1	
Attraction Name To &AttractionNameTo	3	Eiffel Tower	2	1	
GRID Id Attraction Name AttractionName Country Photo CountryName CountryName	Event Load &trips = Count(TripDate) endevent				

O evento Load vai ser executado automaticamente para cada registro

$${}_{\rm Page}24$$

	GeneXus	
Trip TripDate TripDescription Attraction AttractionName	For each <u>Main_Code</u> <u>Atrips = Count(Trip</u> Print Attractions endfor	Date)
	AttractionId AttractionName CountryId CityId	
Attraction Name From &AttractionNameFrom	1 Louvre Museum 2 1	
	2 The Great Wall 3 1	
Attraction Name To &AttractionNameTo	3 Eiffel Tower 2 1	
GRID Id Attraction Name Country Photo Trips AttractionName CountryName CountryName	Event Load &trips = Count(TripDate) endevent	J

da tabela base do grid que cumpra com as condições de filtro, imediatamente antes que seja inserida a linha no grid.



É por essa razão que ao ser executado o seu código sabe-se que estamos trabalhando com um registro da tabela base e sua estendida, e essa fórmula inline:



não vai contar todos os trips



só aqueles da tabela TripAttraction que correspondam a esse AttractionId, o do registro de Attraction que estamos a ponto de carregar no grid.

Page Z (

	Gene	(us ⁻
Trip TripDate TripDescription TripDescription AttractionId AttractionId AttractionName	For each <u>Main_Code</u> <u>&trips = Count</u> Print Attraction endfor	<u>TripDate)</u> s
	AttractionId AttractionName CountryId City	/ld
Attraction Name From &AttractionNameFrom	1 Louvre Museum 2 1	
	2 The Great Wall 3 1	
Attraction Name To &AttractionNameTo	3 Eiffel Tower 2 1	_
GRID Id Attraction Name AttractionName CountryName CountryName CountryName	Event Load &trips = <u>Count(TripDate</u> new endevent line!	,)

Implementemos em GeneXus. Já temos a transação Trip criada. Vamos até a seção de eventos da web panel. Nesse combo são disponibilizados os eventos predefinidos, ou seja, os eventos do sistema que são produzidos em momentos específicos, nos quais poderemos programar código.

WWAttractionsFromScratch * ×	•
Web Form * Rules Events Conditions Variables *	
Enter	
Load	6
OnMessage	
Refresh	
Start	
TrackContext	

Entre eles, temos o evento Load. Ao escolhê-lo, aparece esse código:

l	_	Event	Load
	L	Endeve	ent

Aqui dentro programaremos o que queremos que seja executado a cada vez que se está posicionado em um registro da tabela Attraction, antes de carregar a linha no grid. Em nosso caso...

$${\rm Page}\,27$$

WWAttractionsFromScratch ×	-
Web Form Rules Events Conditions Variables	
Load ~	
1 Event Load	E
2 &trips = count(TripDate)	
	<u>^</u>
	~
<	>

Executemos... Terá que reorganizar a base de dados para adicionar as tabelas correspondentes a nova transação Trip.

Aqui mostramos a execução com um par de trips já cadastrados.

Trip		
	« < > » SELECT	
ld	1	
Date	09/12/16 29	
Description	А	
Attraction		
Attraction Id Attraction	Name	
× 24 😚 Eiffel Towe	r	
× 25 🕆 Christ the	Redemmer	
× 27 🏠 Matisse M	× 27 🔐 Matisse Museum	
0		

Trip	
	« < > » SELECT
ld	2
Date	09/13/16 29
Description	В
Attraction	
Attraction Id Attraction	n Name
× 24 🔓 Eiffel Tow	er
× 25 🕆 Christ the	Redemmer
× 26 🕆 Smithsoni	an Institute
0	



Vemos que a torre Eiffel está em um par de Trips, o Cristo Redentor também, Smithsonian Institute em um, e o museu Matisse também em um.

Agora executemos nossa web panel.



E vemos que está mostrando corretamente o que queríamos.

Agora queremos adicionar a soma de excurções nas quais as atrações que são mostradas no grid em cada oportunidade estão incluídas. Isso é, a soma dos valores dessa coluna:



Para isso, adicionamos uma variável fora do grid:

WWAttractionsFromScratch X	Toolbax	4 ×
Web Form Rules Events Conditions Variables	Controls	-
✓ <no action="" group="" selected=""></no>	⊂™ Attribute/Variable	
	Button	
Im Main Table Height I Imut trips	Embedded Page	
	Error Viewer	
Country Id &CountryId ~	- Horizontal Rule	
	lmage	
Attraction Name From & AttractionNameFrom	A Text Block	=
	Web Component	
Attraction Name To &AttractionNameTo	Containers	
	Free Style Grid	
	E Grid	
Id Attraction Name Country Bosto Tring	XY Group	
AttractionId AttractionName CountryName 8trins	Html	
	Responsive Table	
	Section	
🖾 Attribute/Variable	Tab Tab	
Insert into MainTable	Table	
	Extended Controls	

Colocamos o nome de totalTrips:

Insert Attribute/Va	riable				\times
Filters					
Pattern: New	v Variable		×		
Object Type	Al 🌾 Filter		X		
Object.	/ariable: &totalTrips	5	^		
Nam	Name	totalTrips			^
& count	Description	total Trips			
& Attrac	Column title	total Trips		Name	
& Attrac	Class	Attribute		vame	
a Time		<u>Help</u>			
Pamp	 Type Definition 				
Pgmd	Based on	(none)			
trips	Data Type	Numeric			
88 Airline	Length	4		ount	
88 Airline	Decimals	0			
88 Airline	Signed	False		1e	
88 Airpor	Collection	False			
8 Airpor	Dimensions	Scalar		ne	
& Attrac	Initial value			ddress	
			*	lame	
& Attraction	010	OK Cancel		hoto	
CategoryId	_	Id Catego	ory k	d	¥
New Variable	Edit	ОК		Cancel	

Alteramos a propriedade **Label Caption** para que sua etiqueta apareça em maiúsculas, já que o nome possuímos em minúsculas:

$${}^{\rm Page}30$$

WWAttractionsFr	romScratch * ×			-	0	Properties		4 ×
Neb Form * Rule	es Events Condition	ns Variables *			Ge	neral Class		
<no action="" group<="" td=""><td>selected</td><td></td><td></td><td></td><td>></td><td>Al Filter</td><td></td><td>\times</td></no>	selected				>	Al Filter		\times
					~	Attribute/Variable: &total	Trips	
	total hps			4		Control Name	&totalTrips	
]		Attribute	&totalTrips	
Country Id &C	Countryld ~					Label Position	Left	
						Label Caption	Total Trips	
Attraction Nar	me From &Attractio	nNameFrom				Readonly	False	
						Return On Click	False	
Attraction Nar	me To 8/AttractionN	lameTo				> Appearance		
Attraction Nan	and to contractioning	amero				✓ Control Info		
						Control Type	Edit	
GRID	7	1	1			Input Type	Values	
ld Attractional d	Attraction Name	Country	Photo Trips			Notify Context Change	False	
Attractionid	Attractionivame	Countryivame	octrips			> Behavior		
						> Cell information		
						> Row information		
O ototal Trips &te	otalTrips	0	•					
	sumps	0						
-		-		-				

Um forma eficiente de calcular o valor exibido pelas variáveis é... a cada vez que uma linha for carregada no grid, somar o valor da variável & Trips dessa linha ao valor calculado até o momento em & totalTrips.

No evento Load, após o cálculo do valor da variável &trips, atribuir a variável &totalTrips o valor que contém até o momento mais o valor da variável &trips:

Event Load
<pre>&trips = count(TripDate)</pre>
&totalTrips = &totalTrips + &trips
Endevent

Para a primeira linha a ser carregada no grid, será executado pela primeira vez o evento Load, calcula-se & Trips e quando for calculada & total Trips, o valor atual estará em zero,

			G	eneXu	S
	AttractionId	AttractionName	Countryld	Cityld	
_	1	Louvre Museum	2	1	
	2	The Great Wall	3	1	
Country Id &CountryId ~	3	Eiffel Tower	2	1	
Attraction Name To &AttractionNameTo GRID	Eve & & end	nt Load .trips = Count(1 .totalTrips = &to event	TripDate_) talTrips + 8	∉ &trips	
ld Attraction Name Country Photo Trips AttractionName CountryName CountryName	&	trips 2]		
Total Trips &ttotalTrips	&tota	ITrips 0]		
			1		

assim que nesse caso &totalTrips assumirá o valor de &Trips.

$$_{\text{Page}}31$$

					AttractionId	AttractionName	Countryld	CityId
					1	Louvre Museum	2	1
				1	2	The Great Wall	3	1
Country Id 8	Countryld ~				3	Eiffel Tower	2	1
GRID Id	Attraction Name	Country	Photo	Trips				
	AttractionName	CountryName		8ttrips	8	trips 2]	
AttractionId								

Para a segunda linha a ser carregada, calcula-se o valor de &trips e &totalTrips conterá o valor anterior

					AttractionId	AttractionName	Countryld	CityId
					1	Louvre Museum	2	1
Country Id &	Countryld ~				2	The Great Wall	3	1
, [3	Eiffel Tower	2	1
GRID	1			1	& ende	totalTrips = &to event	tal Trips + 8	ktrips
ld AttractionId	Attraction Name AttractionName	Country CountryName	Photo	Trips &trips	&	trips 1]	

ao que será somado ao valor da variável & trips para essa segunda linha, e assim sucessivamente.

						Ge	eneXus
				AttractionId	AttractionName	Countryld	CityId
				1	Louvre Museum	2	1
Country Id	Countrald			2	The Great Wall	3	1
Country in a	countryia v			3	Eiffel Tower	2	1
GRID	Attraction Name Cou	intry Photo	Trips	& ende	totalTrips = &to event	talTrips + &	trips
AttractionId	AttractionName Con	antryName	8ttrips	&	trips 1]	
Total Trips 8	totalTrips			&total	Trips	1	

Logo que seja carregada a última linha, a variável &totalTrips terá o valor desejado.

 ${}^{\rm Page}32$

			G	eneXu	S
	AttractionId	AttractionName	Countryld	Cityld	
	1	Louvre Museum	2	1	
Sunda Id. BCounteddau	2	The Great Wall	3	1	
	3	Eiffel Tower	2	1	
Attraction Name To &AttractionNameTo	Eve & & ende	nt Load trips = Count(1 totalTrips = &to event	TripDate) talTrips + 8	≪ &trips	J
Attraction Name Country Photo Trips & & & & & & & & & & & & & & & & & & &	&	trips 1]		
otal Trips &totalTrips	&total	Trips 3	1		

Executemos para testar.

WWAttractions From Sci	rat X						1	-		×
← → C 🗋 trialap	ps3.genexus.	com/ld8	562acf4c	97c0fe8095	a5c284d6	88e07/ww	Image: A = 1	2	<u>a</u> 0	Ξ
Applica	tion Na	ame					_{by} G	ene	Xuš	Â
Recents W	WAttractions	From								
Country Id	(None	e)	•							
Attraction Name From										
Attraction Name To										
Id Attraction Name	Country	Photo	Trips							
25 Christ the Redemme	er Brazil		2							
24 Eiffel Tower	France	Å	2							
28 Forbidden city	China		0							
22 Louvre Museum	France	-	0							
27 Matisse Museum	France		1							
26 Smithsonian Institute	e United State	s militaria	1							
23 The Great Wall	China	No.	0							
Total Trips		6								

Podemos ver duas coisas: primeiro, que está somando corretamente; segundo: por tratar-se de uma variável, é de entrada, pelo que o usuário pode alterar o valor, o que não tem sentido. Assim, em primeiro lugar, configuremos a propriedade Readonly.

Para isso nos posicionamos sobre a variável no form e entre suas propriedade, alteramos a ReadOnly para True:

Class Filter ribute/Variable: &totalT ntrol Name ribute el Position el Costion	rips &totalTrips &totalTrips Left TotalTrips
Filter ribute/Variable: &totalT ntrol Name ribute el Position	rips &totalTrips &totalTrips Left TotalTrips
ribute/Variable: &totalT ntrol Name ribute el Position	rips &ttotalTrips &ttotalTrips Left Total Trips
ntrol Name ribute el Position	8xtotalTrips 8xtotalTrips Left Total Trips
ribute el Position	&totalTrips Left Total Trips
el Position	Left Total Trips
al Cantion	Total Trips
caption .	rotar mps
adonly	Тгие
urn On Click	False
ppearance	
Control Info	
Control Type	Edit
nput Type	Values
Jotify Context Change	False
	ippearance Control Info Control Type nput Type Votify Context Change

Pode chamar a nossa atenção, também, que &trips também é uma variável, está sendo exibida como readonly, e porém, não fizemos nada para isso. As variáveis nos grids, quando não se tem programado eventos no nível das linhas, e nem são percorridas com for each line via código, serão readonly. Voltaremos a esse tema logo.

Mas também, vejamos o que passa se filtrarmos, por exemplo, por França.

WWAttractions From Scrat		1	-		×
← → C 🗋 trialapps3.g	enexus.com/ld8562acf4c97c0fe8095a5c284d688e07/	ww e e 숬	2	e 0	Ξ
Applicatio	n Name	by	Ger	eXu:	Ŝ
Recents WWAtte	ractions From				
Country Id	France •				
Attraction Name From					
Attraction Name To					
Id Attraction Name Country	y Photo Trips				
24 Eiffel Tower France	2				
22 Louvre Museum France	0				
27 Matisse Museum France	1				
Total Trips	9				

No lugar de mostrarmos um total de 3, está sendo mostrado 9, que é a soma do valor que era mostrado antes, 6, mais os 3 que agora deveria estar sendo mostrado. Por que aconteceu isso?

Ao alterar uma das variáveis de filtro, a web panel voltou a carregar o grid. Isso é, voltou a consultar a tabela

 $_{Page}34$

Attraction da base de dados, e voltou a executar o evento **Load** para cada registro que cumpra os filtros. O problema é que a variável &totalTrips teria que ser zerada, antes de lançar a carga do grid.

Onde fazemos isso? No evento Refresh.

Esse evento do sistema é produzido sempre que a web panel é carregada, imediatamente antes de ir a base de dados buscar a informação para carregar no grid. Isso é, imediatamente antes de executar os eventos Load para cada linha a ser carregada.

Assim, vamos até a aba de eventos, e escolhemos no combo o Refresh...

wwAttractionsFromScrat	ch ×	*
Web Form Rules Events	Conditions Variables	
Load 🗸		
Enter		E
OnMessage	= count(TripDate)	
Refresh	rips = &totalTrips + &trips	~
Start		
TrackContext		
		\sim
<		>

Esse é o momento para colocar zero na variável.

WWAttractionsFromScratch * ×	-
Web Form Rules Events* Conditions Variables	
Refresh ~	
1 Event Load	1
<pre>2 &trips = count(TripDate)</pre>	
3 &totalTrips = &totalTrips + &trips	
4 - Endevent	
5	
6 ⊒ Event Refresh	
7 &totalTrips = 0	
8 – Endevent	
	`

Observemos que a ordem em que os eventos ficam escritos não tem a menor importância. Aqui somente indica-se o código que será executado ao produzir cada um deles.

$${}^{\rm Page}35$$

Pressionemos F5.

Podemos ver que agora o total de trips aparece Readonly. Ao executar esta web panel pela primeira vez, são disparados três eventos de forma consecutiva: o **evento Start**, que executará somente ao ser aberta a web panel, na primeira vez, o **evento Refresh**, que põe a variável em zero, e o **evento Load**, tantas vezes forem as linhas carregadas no grid. Nesse caso foram 7.

WWAttractions From Scra	×				±	-		×
← → C 🗋 trialapp	os3.genexus.	com/Id85	62acf4c9	7c0fe8095a5c284d688e07/wwattr	ac 🗟 🖒	2	 0	Ξ
Applicat	tion Na	ame			_{by} G	iene	eXuš	
Recents W	WAttractions	From						
Country Id	(No	ne)	۳					
Attraction Name From								
Attraction Name To								
				Start (only the first time)				
Id Attraction Name	Country	Photo	Tripe	Refresh (once)				
25 Christ the Redemme	r Brazil		2	No. of the second se				
24 Eiffel Tower	France	Å	2	Load (7 times)				
28 Forbidden city	China		0					
22 Louvre Museum	France		0					
27 Matisse Museum	France		1					
26 Smithsonian Institute	United State	s miñie	1					
23 The Great Wall	China	XXX	0	1				
Total Trips	6							•

Agora, se escolhemos filtrar por um país, por exemplo, França, visualizamos que está sendo calculado bem o número de trips.

Ao mudar o valor de uma variável que tem efeitos sobre as condições que devem cumprir os registros para carregar-se no grid, volta-se a disparar o evento **Refresh** (e portanto a variável &totalTrips volta a ficar com zero) e a ida a base de dados para voltar a filtrar e carregar os registro no grid. Portanto o **Load** volta a ser disparado para cada atração da França a ser carregada.



WWAttractions From Scrall X	*	-		×
← → C 🗋 trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwa	8 S	2	a 0	≡
Application Name	by	Ge	neXu	Ŝ
Recents WWAttractions From				
Country Id France				
Attraction Name From				
Attraction Name To				
Id Attraction Name Country Photo Trips				
24 Eiffel Tower France 2 Refresh (once) Load (3 times)				
22 Louvre Museum France 0				
27 Matisse Museum France				
Total Trips 3				

Se agora escolhemos ver as atrações entre E e F, também volta-se a atualizar a tela. Volta-se a colocar em zero a variável &totalTrips no Refresh, e volta-se a ler a tabela base do grid e sua estendida, desta vez recuperando apenas um só registro, pelo que o Load será executado uma única vez.

 ✓ WWAttractions From Scralx ← → C	enexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwa	- •	2	_ 🔒 ()	×
Applicatio	n Name		Ger	neXu	Ŝ
Recents WWAttr	actions From				
Country Id	France				
Attraction Name From	E				
Attraction Name To	F				
Id Attraction Name Country	Photo Tripe Refresh (once)				
24 Eiffel Tower France	2 Load (once)				
Total Trips	2				



Agora bem, se observarmos a web panel que temos implementada até o momento, poderemos apreciar que vai aparecendo o objeto que havia criado o Pattern WorkWith aplicado na transação Attraction.

Esse objeto era, evidentemente, uma web panel.

Attractions	;	×						± _	D X
ତ	trialapps:	3.genexus.com/ld8562a	cf4c97c0fe8095a5	c284d688e07/wwa	attraction.aspx			ඉ ☆ ඛ by Gen e	<mark>e 0</mark> ≡
Rece	FILTERS	Attractions	Q Name					+ IN3	SERT
	Id	Name	Country Name	Category Name	Photo	City Name			
	25	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE	
	24	Eiffel Tower	France	Monument	Å	Paris	UPDATE	DELETE	
	28	Forbidden city	China	Tourist Site		Beijing	UPDATE	DELETE	
	22	Louvre Museum	France	Museum		Paris	UPDATE	DELETE	
	27	Matisse Museum	France	Museum		Nice	UPDATE	DELETE	
	26	Smithsonian Institute	United States	Museum	antitus.	Washington	UPDATE	DELETE	
	23	The Great Wall	China	Tourist Site	**	Beijing	UPDATE	DELETE	

Aqui filtrava-se por uma única variável:

Attraction	5	×						<u> </u>	
ארץ → פ ארץ	C I trialapps3.genexus.com//d8562acf4c97c0fe8095a5c284d688e07/wwattraction.aspx								
Rece	ents Attractio	ns							
✓ SHOW	FILTERS	Attractions	Q Name					+ IN	SERT
	Id	Name	Country Name	Category Name	Photo	City Name			
	25	Christ the Redemmer	Brazil	Monument		Rio de Janeiro	UPDATE	DELETE	
	24	Eiffel Tower	France	Monument	A	Paris	UPDATE	DELETE	
	28	Forbidden city	China	Tourist Site		Beijing	UPDATE	DELETE	
	22	Louvre Museum	France	Museum		Paris	UPDATE	DELETE	
	27	Matisse Museum	France	Museum		Nice	UPDATE	DELETE	
	26	Smithsonian Institute	United States	Museum	<u>milin</u>	Washington	UPDATE	DELETE	
	23	The Great Wall	China	Tourist Site		Beijing	UPDATE	DELETE	



Mas, o mais interessante é que além de permitir filtrar os dados do grid, o Work With oferece executar ações

sobre os dados. Por exemplo, poder atualizar os dados de uma atração ou eliminar a atração assim como inserir uma nova.

Para isso o pattern inseriu dois controles no nível das linhas do grid, e um fora. Em qualquer dos três casos, a ação asosciada a cada controle consiste em chamar a transação Attraction, enviando como parâmetro o **modo** em que deve ser aberta a transação, isto é, se é chamada para atualizar, eliminar ou inserir. Nos dois primeiros, Update ou Delete como corresponderão a eventos de uma linha contaremos com o id da atração da linha que será enviado como segundo parâmetro na transação, de modo a atualizar os dados **dessa** atração, ou eliminar **essa** atração. No caso do controle Insert fora do grid, será enviado **0** como segundo parâmetro, já que as atrações no Insert são autonumeradas.

É por isso que o pattern alterou a transação Attraction, adicionando, entre outras coisas, a regra Parm:



que como vemos, recebe duas variáveis: a variável & Mode é uma variável padrão nas transações, Character de 3:

wwAttractionsFromScrate	h 🗙 💽 WWAttraction 🗙 🗐	Attraction \times		•
Structure 🐔 Web Form R	ules Events Variables Patterns			
Name	Туре	Is Collection	Description	^
⊡·& Variables				
🖹 🚷 Standard Variables				
 GxRemove 	Numeric(1.0)		Gx Remove	_
• Mode	Character(3)			
···· Pgmdesc	Character(256)		Pgmdesc	
···· Pgmname	Character(128)		Pgmname	
···· • Time	Character(8)		Time	
• Today	Date		Today	
 AttractionId 	Attribute:AttractionId		Attraction Id	
 Insert_CategoryId 	Attribute:CategoryId		Insert_Category Id	
··· • Insert_CityId	Attribute:CityId		Insert_City Id	
···· Insert_CountryId	Attribute:CountryId		Insert_Country Id	
 Insert_SupplierId 	Attribute:SupplierId		Insert_Supplier Id	
 IsAuthorized 	Boolean		Is Authorized	
··· • TrnContext	TransactionContext		Trn Context	
• TrnContextAtt	TransactionContext.Attribute		Trn Context Att	¥

que aceita um de quatro valores especificados no domínio enumerado TrnMode:

- Insert, INS
- Update, UPD
- Delete, DLT
- Display, DSP

Properties	7
📱 🛃 🌾 Filter	>
Domain: TrnMode	
Name	TrnMode
Description	TrnMode
Empty as null	Yes
Class	Attribute
Module	GeneXus
Qualified Name	GeneXus.TrnMode
Object Visibility	Public
v Type Definition	
Based on	(none)
Data Type	Character
Length	3
Enum Values	Insert, Insert (INS) Update, Update (UPD) Delete, Delete (DLT) Display, Display (DSP)
Collection	False
Initial value	
Enable national language support	No

Recebendo um desses quatro valores, a transação saberá em que modo deve ser aberta.

E por outro lado, receberá como segundo parâmetro o id da atração, na variável & AttractionId, para quando se quer fazer update, delete ou display.

🐼 WWAttractionsFromScratch X 😨 WWAttraction X 🗐 Attraction X	•
Structure 🖌 Web Form Rules Events Variables Patterns	
<pre>1 /* Generated by Work With Pattern [Start] - Do not change // 2 [web] 3 { 4 parm(in:&Mode, in:&AttractionId); 5 6 AttractionId = &AttractionId if not &AttractionId.IsEmpty(); 7 noaccept(AttractionId); </pre>	II II
<pre>8 noprompt(AttractionId); 9 10 CountryId = &Insert_CountryId if &Mode = TrnMode.Insert and not &Insert_CountryId.IsEmpty(); 11 noaccept(CountryId) if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty(); 12 CategoryId = &Insert_CategoryId if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty(); 13 noaccept(CategoryId) if &Mode = TrnMode.Insert and not &Insert_CategoryId.IsEmpty(); 14 CityId = &Insert_CityId if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty(); 15 noaccept(CityId) if &Mode = TrnMode.Insert and not &Insert_CityId.IsEmpty(); 16 /* Generated by Work With Pattern [End] - Do not change */ 17</pre>	~
	>

Em nossa web panel implementaresmo uma dessas ações sobre as atrações. Por exemplo, a de Update. A intenção é mostrar um exemplo de ações sobre os dados.

Teremos que inserir um controle no grid. No caso do pattern é inserida uma variável character chamada update,

$$_{\rm Page}40$$

WWAttractionsFromScratch × 😨 WW	Attraction ×			•
Actions Insert EmainTable EmainTable EmainTable	id 💷 Update			•
Hide Filters	Attractions	<actions></actions>	&AttractionNam	e
Ordered By <	rorViewer: ErrorViewer>			
Name Country GRI	D	· · · · · · · · · · · · · · · · · · ·		-0
	Name tractionId AttractionName	Country Name Category Nam	ne Photo City Name	Rilladata RiDalata
Country Name				
&CountryName				
<				>

🔜 WWAttractionsFromScratch 🗙 📑 W	WAttraction $ imes$
🐔 Web Form 🛛 🐔 Rules 🛛 🐔 Events 🛛 🐔 C	Conditions <u>5</u> Variables
Name	Туре
🖃 🚷 Variables	
🗄 🚷 Standard Variables	
ADVANCED_LABEL_TEMPLATE	Character(20)
 AttractionName 	Attribute:AttractionName
 CountryName 	Attribute:CountryName
 Delete 	Character(20)
 GridPageCount 	Numeric(8.0-)
 GridState 	GridState
 GridStateFilterValue 	GridState.FilterValue
HTTPRequest	HttpRequest
 IsAuthorized 	Boolean
 OrderedBy 	Numeric(4.0)
 Session 	WebSession
TrnContext	TransactionContext
TrnContextAtt	TransactionContext.Attribute
Update	Character(20)

e a que se atribui o texto "UPDATE" que visualizamos em execução. Mas nós escolhemos inserir uma imagem, que antes de qualquer coisa devemos inserir na KB.

🛃 updatelcon 🗙			
Images			
🛛 🔯 New Image			
Image	Theme	Language	Density
updatelcon24.png (Default) Size:24x24 px	Any	Any	Unknown
- Preview	•	•	·

 $_{\rm Page}41$

Chamamos a imagem de updatelcon.

Agora vamos na web panel e arrastamos da toolbox o controle Attribute/Variable até a última coluna do grid, definimos a nova variável como &Update, mas não do tipo character, e sim do tipo Image:

🐼 Start Page X 🔜 WWAttractionsFromScratch* X 🛅 Navigation View X	O Properties	. 4 ×
Web Form * Rules Events Conditions Variables	General Class	
<no action="" group="" selected=""></no>	🔠 🦺 🌾 Filter	\times
	Attribute/Variable: &Update	
Main lable E Grid Dpdate	Control Name &Update	
	Attribute &Update	
Country Id &CountryId ~	Title	
	Class Inage	
Attraction Name From & Attraction NameFrom	Column Class	
	Beturn On Click False	
	On Click Event	
Attraction Name 10 CAttractionName10	> Control Info	
	> Behavior	
GRID	✓ Appearance	
Id Attraction Name Country Photo Trips Update	Auto Resize True	
AttractionId AttractionName CountryName & Krips	Visible True	
	Alternate Tex	
	Tooltip Text	
rotar mps eccotar mps		
< >		

Retiramos o título para que não figure como título da coluna, e nos resta carregar essa variável com a imagem que acabamos de inserir na KB. Onde fazemos?

Se a imagem fosse variar por linha do grid, faríamos no evento Load, mas a imagem será a mesma para cada linha, e não irá variar nunca, assim que uma boa opção é fazer no **evento Start**, que será executado uma única vez, quando a web panel é aberta, e não mais.

Assim que inserirmo no evento Start, e ali escrevemos:



Page4

E como parâmetro do método, escrevemos o nome da imagem da KB, isto é, updateIcon.

🐼 Start Page 🗙 🔜 WW/AttractionsFromScratch* X 🗐 Navigation View X	•
Web Form * Rules Events * Conditions Variables	
Start ~	
1 🗆 Event Load	
<pre>2 &trips = count(TripDate)</pre>	3
3 &totalTrips = &totalTrips + &trips	=
4 – Endevent	
5	
6 🖂 Event Refresh	
7 &totalTrips = 0	
8 - Endevent	
9	
10 Event Start	
11 &Update.FromImage(updateIcon)	
12 Endevent	

Vejamos em execução.

WWAttractions From Scra	×	≜ – □ ×
← → C 🗋 trialapp	os3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07	/wwat 🗟 🖧 🕤 🤬 🚺 😑
Applicat	ion Name	by GeneXuš
Recents W	WAttractions From	
Country Id	(None)	
Attraction Name From		
Attraction Name To		
Id Attraction Name	Country Photo Trips	
25 Christ the Redemme	r Brazil 2 🖍	
24 Eiffel Tower	France 2 🖍	
28 Forbidden city	China 💓 0 🖍	
22 Louvre Museum	France 0 🖍	
27 Matisse Museum	France 1 🖍	
26 Smithsonian Institute	United States 🚂 1 🖍	
23 The Great Wall	China 💦 0 🖍	
Total Trips	6	

Agora nos resta associar um evento a essa imagem, de tal maneira que quando o usuário clique sobre ela, produza esse evento e execute o seu código, no qual chamaremos a transação Attraction.

 $_{\rm Page}43$

Há várias alternativas para realizar isso. Uma delas é ir até a aba de eventos e com Insert/Event vemos à

esquerda todos os nomes dos controles que temos inseridos no form. Escolhemos o que desejamos, a variável &Update, e vemos que à direita é mostrado os eventos que podem ser associados, por exemplo, o evento click:



Desse modo, quando o usuário clique sobre a imagem para uma linha, executará o código que escrevemos dentro desse evento. O que queremos fazer nesse caso é chamar a transação Attraction:

 $_{
m Page}44$

🐼 Start Page X 🔜 WWAttractionsFromScratch * X 🗐 Navigation View X
Web Form * Rules Events * Conditions Variables
&Update.Click ~
1 Event Load
<pre>2 &trips = count(TripDate)</pre>
3 &totalTrips = &totalTrips + &trips
4 - Endevent
5
6 ⊡ Event Refresh
7 &totalTrips = 0
8 - Endevent
9
10 Event Start
11 &Update.FromImage(updateIcon)
12 - Endevent
13
14 Event &Update.Click
15 Attraction(
16 - Ende Call(Mode: Character, AttractionId: Numeric)
<

Passando o modo Update, isto é, o valor Update do domínio enumerado TrnMode que vimos antes, e o valor de AttractionId correspondente a linha do grid onde foi clicada:



Executemos para testar.



WWAttractions From Scr	at X							*	-		×
🗲 🔶 C 🗋 trialap	ps3.genexus.c	om/Id8562	acf4c97c0)fe809)5a5c28	4d688e	e07/wwat	e 1	2	<u>e</u> 0	≡
Applica	tion Na	ime							Gei	neXu	Ŝ
Recents A	ttractions — W	VWAttractio	ns From								
Country Id	(None	e) 🔻]								
Attraction Name From											
Attraction Name To											
Id Attraction Name	Country	Photo	Trips								
25 Christ the Redemme	er Brazil	1	2								
24 Eiffel Tower	France		2								
28 Forbidden city	China		0								
22 Louvre Museum	France		0								
27 Matisse Museum	France		1								
26 Smithsonian Institute	e United States	<u>miâte</u>	1								
23 The Great Wall	China		0								
Total Trips	6										

Primeiro observamos que a coluna da variável & Trips aparece agora como editável. Não havíamos definido como Readonly explicitamente porque ao executar vimos que já estava. Como falamos antes, as variáveis do grid em princípio são colocadas como readonly, exceção quando é definido algum evento no nível das linhas, como é o nosso caso, ou outras exceções que agora não veremos.

Configuremos como Readonly. Voltamos a executar.



Selecionamos, por exemplo, o país França:



WWAttractions From Scrall X	*	-		×
← → C Trialapps3.genexus.com/ld8562acf4c97c0fe8095a5c284d688e07/wwat	R 🖒	2	e 0	Ξ
Application Name		Gei	neXu:	Ŝ
Recents Attraction — WWAttractions From				
Country Id France •				
Attraction Name From				
Attraction Name To				
Id Attraction Name Country Photo Trips				
24 Eiffel Tower France 2				
22 Louvre Museum France 0				
27 Matisse Museum France 📰 1 🖍				
Total Trips 3				

E agora clicamos sobre a imagem de update para a Torre Eiffel:

Attraction ← ⇒ C [*] □ tria	× alapps3.genexus.c	om/ld8562acf4c97c0fe8095a5c284d688e07/attra	<u>ب</u> برها کې	2	_ 	×
Applic	cation Na	me	_{by} G	iene	Xuš	
Recents	WWAttractions F	From — Attraction				
	Attraction					ł
	Id	24				l
	Name	Eiffel Tower]			l
	Country Id	2				l
	Country Name	France				l
	Category Id	11 💡				ľ
	Category Name	Monument				
	Photo					

Vemos como é chamada a transação em update. Alteremos algo... por exemplo, passamos de maiúsucla a minúscula a letra T de Tower:

 $_{\rm Page}47$

Attraction	×		1	-		×
← → C 🗋 tria	lapps3.genexus.co	m/ld8562acf4c97c0fe8095a5c284d688e07/att	rac 📴 🏠	ລີ (e 0	≡
Applic	cation Na	me	_{by} G	iene	Xus	
Recents	WWAttractions F	rom — Attraction				
	Attraction					I
	ld	24				I
	Name	Eiffel tower				l
	Country Id	2				

Confirmamos... e como o pattern Work With, embora não o vimos, adicionou na transação o comando Return, retorno, para voltar ao objeto que a chamou, neste caso a web panel. Este comando Return é como chamar a web panel pela primeir vez, pelo que será executado o evento Start, seguido do Refresh e do Load tantas vezes quantos registros forem carregados:



É por isso que vemos que ao voltar são carregadas todas as atrações, sem filtros.

Que passaria se não tivéssemos colocado o atributo AttractionId no grid? Ao clicar sobre a imagem para atualizar, que AttractionId seria enviado como parâmetro a transação? Não teria esse valor para enviar.

```
Event &Update.Click
Attraction( TrnMode.Update, AttractionId )
Endevent
```

$$_{\text{Page}}48$$

Como vamos usar um evento no nível das linhas, que será disparado depois que as linhas tenham sido carregadas, não podemos remover o AttractionId do grid. É que aqui já não se está mais na base de dados. O grid armazenou ao carregar-se com o Load todos os valores de suas colunas e nada mais. Um evento posterior trabalhará unicamente sobre os dados carregados no grid. Então, o que podemos fazer se não queremos ver essa coluna no grid é ocultá-la. Continuará estando presente, mas invisível.

🐼 Start Page X 🔜 WWAttractionsFromScratch* X 📋 Navigation View X 🗐 Attraction X	Properties	4 ×
Web Form * Rules Events * Conditions Variables	General Class	
No action group selected>	📜 🤌 🌾 🛛 Filter	×
	Attribute/Variable: A	ttractionId
Main Table	Attribute	AttractionId
	Title	ld
Country Id & CountryId V	Class	Attribute
	Column Class	
Attraction Name From & AttractionNameFrom	Return On Click	False
	On Click Event	
	✓ Control Info	
	Control Type	Edit
	Input Type	Values
Q ^{GRID} Q ····· ▲ Q	Notify Context Cha	r False
Id Attraction Name Country Photo Trips	✓ Behavior	
O AttractionName CountryName Qtrips	Input History	True
	Is Password	False
	Read Only	True
Total Tring Ritotal time	Empty as null	Yes
	✓ Appearance	
	Auto Resize	True
	Format	Text
	Visible	False ~
	Tooltip Text	
	Invite Message	
Show: Build Y X Find:	Brenetties Taalba	
	Properties 100lbc	20

Para isso utilizamos a propriedade Visible com o valor False:

Executamos.

No próximo vídeo veremos o que acontece com um evento no nível das linhas que altera o valor de uma variável do grid. Veremos uma web panel sem grid mas com atributos no form. Faremos um resumo conceitual de tudo o que foi visto, veremos web panels sem tabela base, e mencionaremos mais casos avançados de uso de web panels e suas características.



Page 4