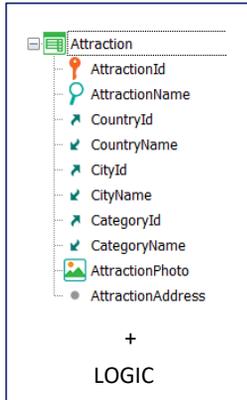
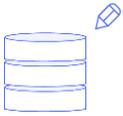


Atualização da base de dados

Usando Business Components. Um exemplo

GeneXus[™]

Business Component



&attraction

AttractionId	7
AttractionName	Forbidden city
CountryId	3
CountryName	China
CityId	1
CityName	Beijing
CategoryId	2
CategoryName	Monument
AttractionPhoto	
AttractionAddress	

+

```

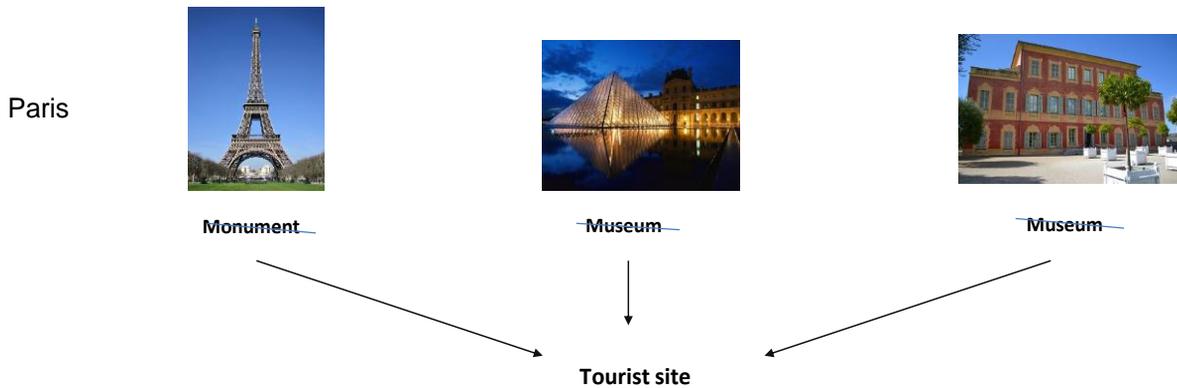
&attraction.Load(Pk)
&attraction.Insert()
&attraction.Update()
&attraction.Delete()

&attraction.Success()
&attraction.Fail()
&attraction.GetMessages()

&attraction.Mode()
&attraction.Save()
&attraction.InsertOrUpdate()
  
```

No vídeo anterior, vimos como inserir, modificar e excluir um registro a partir de um business component correspondente a uma transação de um nível.

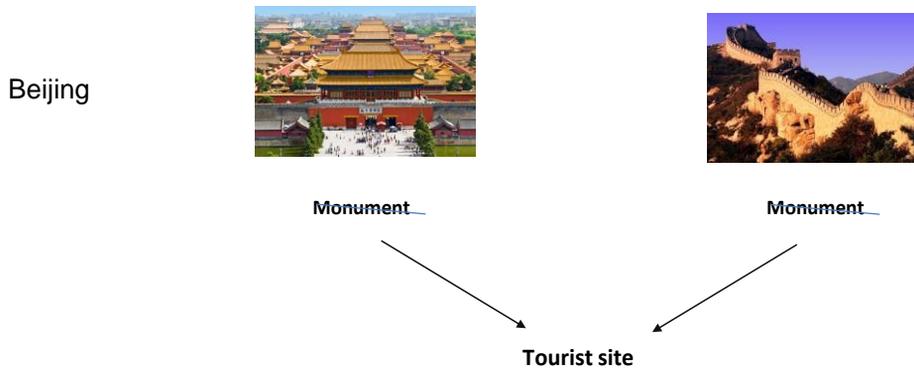
Backoffice new requirement



Agora será fácil para nós implementar o seguinte requisito para o back-office da agência de viagens:

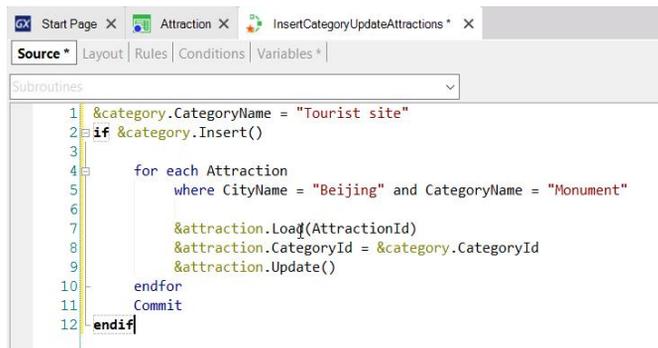
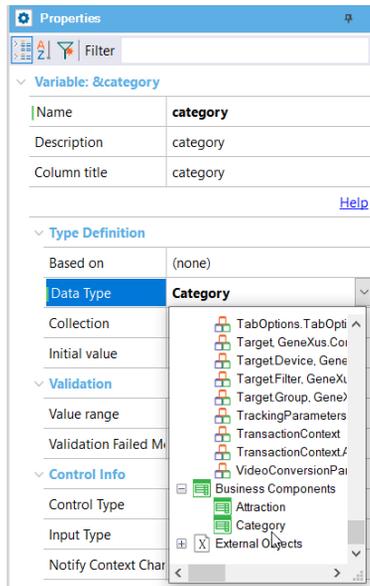
Suponhamos que, em um dado momento, surge a necessidade de criar uma categoria especial, “Tourist site”, que agrupe as atrações turísticas muito importantes de uma cidade, independentemente de que, até o momento, estivessem categorizadas como monumentos, museus, etc.

Backoffice new requirement



Comecemos por algo um pouco mais fácil: suponhamos que na agência, necessitam mudar todas as atrações de Beijing, que antes eram do tipo Monument, para que agora sejam desta nova categoria. Nos pedem para fazê-lo por código, para não termos que fazê-lo de forma interativa, uma a uma através da tela de work with (aqui só vemos duas atrações de Beijing, mas poderia haver muitas mais e, além disso, o usuário poderia deixar passar alguma).

Implementation



Utilizaremos o procedimento que vimos. Apagamos tudo o que havíamos codificado, para começar de novo.

Como teremos que inserir uma nova categoria, devemos criar seu Business Component. E agora definir uma variável desse tipo de dados. Como o Id é autonumerado, não lhe damos valor, e sim o atribuímos ao elemento CategoryName e inserimos.

Se tudo correu bem, então o próximo passo será modificar a categoria de todas as atrações turísticas de Beijing que até agora eram monumentos.

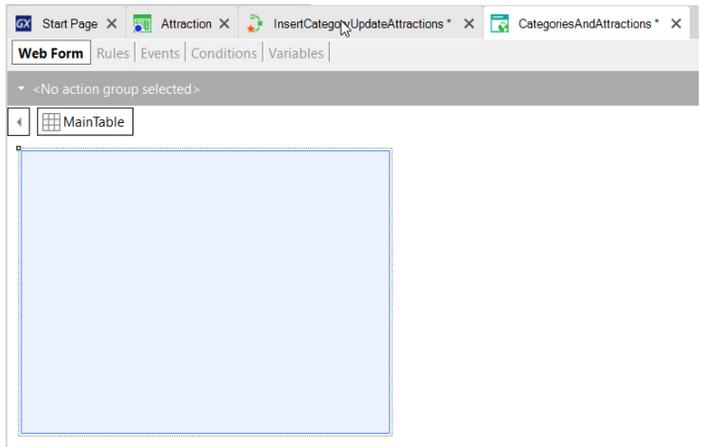
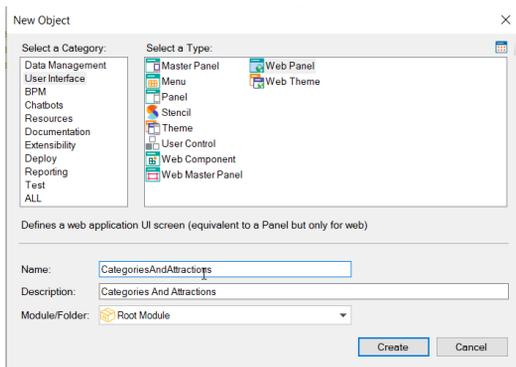
Então devemos percorrer essas atrações com um for each, onde a cidade seja Beijing e a categoria Monument.

Em cada iteração do for each estaremos posicionados sobre uma atração dessas, e o que fazemos é carregá-la na variável &attraction, trocar a categoria pela nova (observemos que aqui estamos recuperando o valor que lhe deu a base de dados ao inserir a categoria, o Id autonumerado). E por último, o que fazemos é modificar o registro na base de dados, disparando todas as regras da transação, como já vimos.

Por fim, escrevemos um Commit.

Vamos invocar este procedimento a partir de uma tela, por isso, removemos a regra outputfile e deixamos as propriedades Call protocol e Main program com os seus valores default.

Web panel

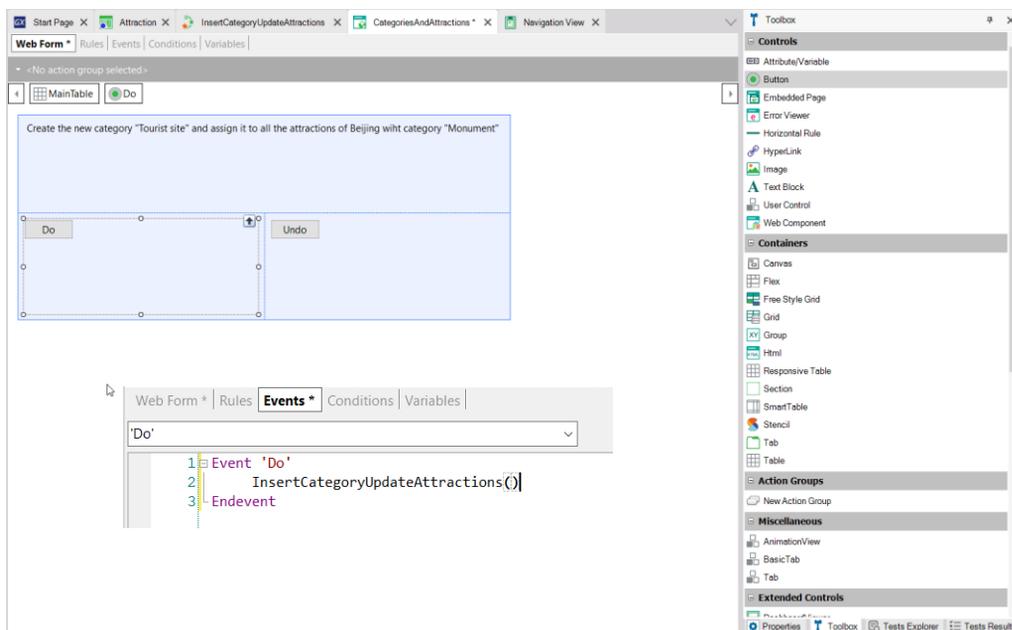


Vamos criar essa tela, esse painel. Será um objeto, do qual falaremos mais adiante, de tipo Web panel, painel web.

Vamos chamá-lo assim...

Aqui vemos o seu web form, vazio. Isto será o que o usuário verá na tela em execução, no seu browser.

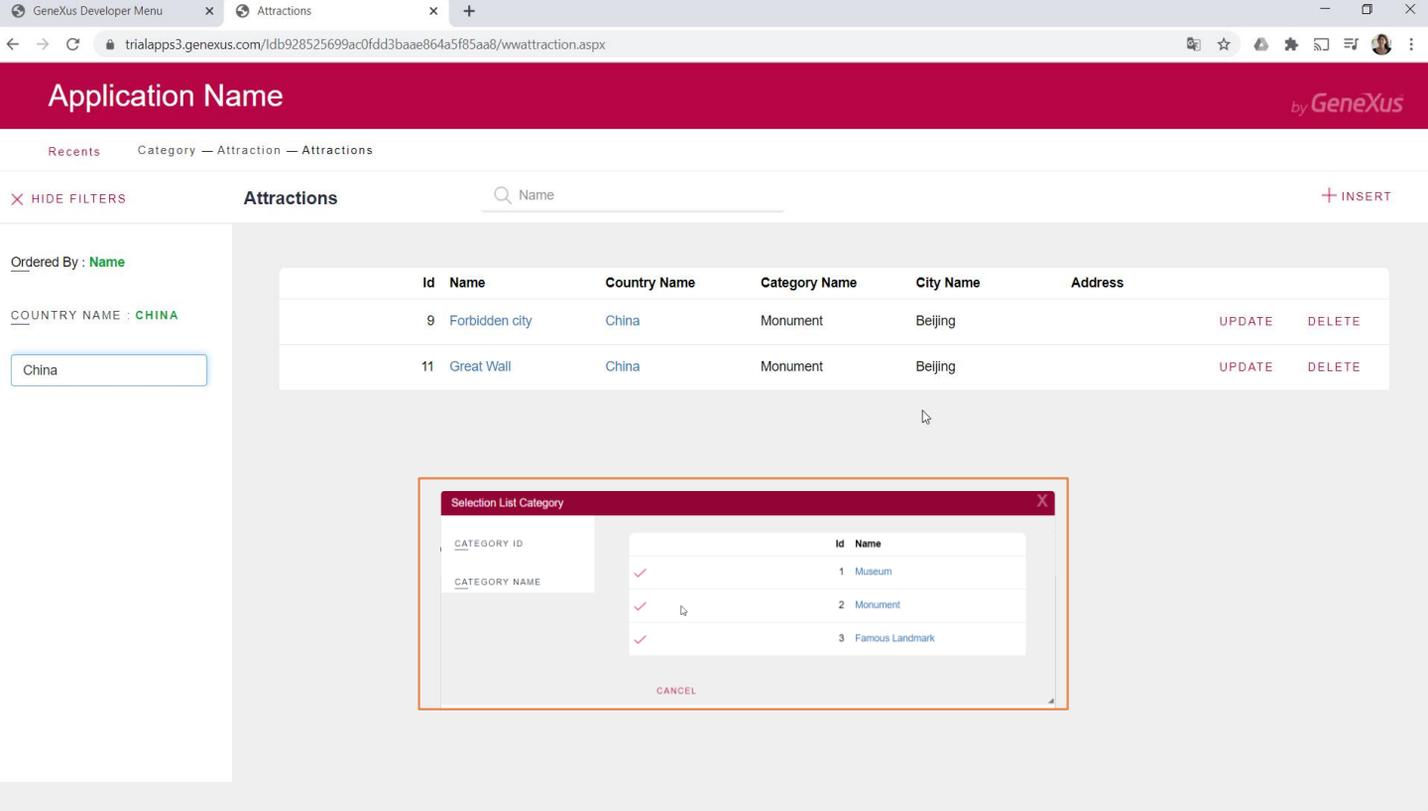
Web panel



Arrastamos a partir da Toolbox um controle do tipo bloco de texto, para mostrar ao usuário uma mensagem, um texto, no painel. Colocamos este Caption, indicando que o que será feito, é inserir a categoria “Tourist site” e modificar as atrações de Beijing. E arrastamos um botão, que terá um evento associado com este nome Do. E vamos colocar outro botão para desfazer a ação.

Queremos programar o código que será executado quando o usuário pressionar o Do. Então vemos que o web panel tem uma seção de eventos, que é onde vamos especificar esse código. O que vamos fazer aqui é invocar o procedimento que tínhamos criado. Podemos digitá-lo diretamente, ou melhor ainda, para evitar erros, arrastá-lo do KB Explorer. E colocar os parênteses.

Vamos testar. F5.



Vejamos as categorias que temos inseridas até o momento na base de dados. São estas três.

Agora vejamos as atrações turísticas. Vamos reinserir a Muralha da China, que tínhamos eliminado no vídeo anterior.

Se filtramos por país, China, vemos que temos aqui as duas atrações de Beijing, que são do tipo Monument. Portanto, deverão ser alteradas, quando executarmos o código do evento do Web panel, pela nova categoria Tourist Site.

Categories And Attractions x Attractions x +

Application Name by GeneXus

Recents Category — Attraction — Categories And Att... — Attractions

Attractions + INSERT

Ordered By: Name

COUNTRY NAME: CHINA

Id	Name	Country Name	Category Name	City Name	Address		
9	Forbidden city	China	Tourist site	Beijing		UPDATE	DELETE
11	Great Wall	China	Tourist site	Beijing		UPDATE	DELETE

Selection List Category X

CATEGORY ID	CATEGORY NAME	Id	Name
		✓	1 Museum
		✓	2 Monument
		✓	3 Famous Landmark
		✓	4 Tourist site

CANCEL

Atualizamos a tela. E vemos que foi alterada efetivamente a categoria. Tendo a criado previamente na tabela.

```
TravelAgency - GeneXus Trial
File Edit View Layout Insert Build Knowledge Manager Window Tools Test Help
[Icons] [Tools] [Release]
KB Explorer
Open: [Name or Pattern]
TravelAgency
  Root Module
  GeneXus
  Airline
  Airport
  Attraction
  AttractionList
  AttractionsByName
  AttractionsPerCategory
  AttractionsReport
  AttractionWithoutParameters
  CategoriesAndAttractions
  CategoriesAndTheirAttractionsList
  Category
  CountriesRanking
  CountriesWithMoreThan2Attractions
  Country
  Customer
  DataProviderCountries
  Diagram1
  Diagram2
  Diagram3
  Diagram4
  Diagram5
  EnterAttractionsFilter
  Flight
  FlightArrivalAirport
  FlightDepartureAirport
  Gx00A0
  Gx00B1
  Gx0010
  Gx0020
  Gx0030
  Gx0040
Output
[Models] [TrialTrunk] [TravelAgency]
INS
Start Page | Attraction | InsertCategoryUpdateAttractions | CategoriesAndAttractions
Web Form | Rules | Events | Conditions | Variables
'Undo'
1 Event 'Do'
2 // InsertCategoryUpdateAttractions()
3 &category.CategoryName = "Tourist site"
4 if &category.Insert()
5
6     for each Attraction
7         where CityName = "Beijing" and CategoryName = "Monument"
8
9             &attraction.Load(AttractionId)
10            &attraction.CategoryId = &category.CategoryId
11            &attraction.Update()
12
13        endfor
14        Commit
15    endif
16 -Endevent
17
18 Event 'Undo'
19 for each Attraction
20     where CityName = "Beijing" and CategoryName = "Tourist Site"
21     &attraction.Load(AttractionId)
22     &attraction.CategoryId = find( CategoryId, CategoryName = "Monument" )
23     &attraction.Update()
24 endfor
25 &categoryId = find( CategoryId, CategoryName = "Tourist site" )
26 &category.Load(&categoryId)
27 &category.Delete()
28
29 if &category.Success()
30     Commit
31 else
32     Rollback
33 endif
34
35 -Endevent
Properties
Filter
Properties | Toolbox | Tests Explorer | Tests Results
```

Como mencionamos no vídeo anterior, as atualizações com business components podem ser realizadas praticamente em qualquer objeto que admita código. Assim, poderíamos ter escrito diretamente aqui o código que tínhamos no procedimento, sem necessidade de invocar qualquer outro objeto. Definimos as variáveis, simplesmente, category e attraction, business components, e pronto.

Vamos programar o código associado ao evento Undo, que o que irá querer fazer é, desfazer as operações anteriores. Ou seja, eliminar a categoria Tourist site, e ao mesmo tempo voltar a deixar as atrações de Beijing, que eram agora Tourist Site com a categoria Monument.

Assim ficaria o código. Estamos percorrendo com um for each as atrações de Beijing de categoria Tourist Site. Estamos carregando a variável attraction, business component, com cada atração encontrada, e modificando seu id de categoria para que corresponda a Monument. Estamos dando a ordem para atualizar.

E depois, vamos eliminar a categoria. Definimos uma variável, CategoryId, para recuperar o identificador da categoria Tourist site, para poder carregar essa categoria no Business component e eliminá-la.

Se tudo correu bem, então realizamos um Commit, e em caso contrário, um Rollback.

Por que definimos esta ordem, ou seja, primeiro percorrer as atrações para depois eliminar a categoria, e não o contrário? O que aconteceria se quiséssemos primeiro eliminar a categoria e depois modificar essa mesma categoria das atrações turísticas? Evidentemente não nos permitirá. Falhará a integridade

referencial, porque estaremos tentando eliminar uma categoria que tem registros associados. Então, a ordem não pode ser o inverso.

Vamos testar. Pressionemos Undo. Vemos que a categoria foi efetivamente eliminada. E se formos às atrações e filtrarmos por China, vemos que restaurou a categoria Monument para as duas atrações.

The screenshot shows the GeneXus IDE interface. At the top, there are several tabs: 'Start Page', 'Attraction', 'InsertCategoryUpdateAttractions', 'CategoriesAndAttractions', and 'MassiveInsertRemove*'. Below the tabs, there's a 'Web Form *' section with sub-tabs for 'Rules', 'Events', 'Conditions', and 'Variables'. The main workspace is divided into two parts. On the left, a 'MainTable' is visible, and a 'RemoveData' button is being dragged from a palette onto the table. On the right, the 'Events' tab is active, showing the code for the 'RemoveData' event:

```

1 Event 'RemoveData'
2   For each Attraction
3     &attraction.Load(AttractionId)
4     &attraction.Delete()
5   endfor
6   For each Category
7     &category.Load(CategoryId)
8     &category.Delete()
9   endfor
10  Commit
11 Endevent

```

Agora vamos criar outro painel para remover todas as informações da tabela de categorias e todas as informações da tabela de atrações turísticas.

Mais uma vez nos aparece o web form vazio. Arrastamos um botão, associamos o evento RemoveData e o codificamos desta maneira.

Outra vez a ordem tem que ser esta. Não poderíamos percorrer as categorias e ir eliminando-as, para, logo depois, passar a eliminar as atrações. Falharia a integridade referencial.

Vamos testar. Pressionamos Remove Data. E agora, se vamos ver a informação na transação Category, vemos que está vazia. O mesmo se formos ver as atrações.

InsertCategoryUpdateAttractions X CategoriesAndAttractions X MassiveInsertRemove X Navigation View X Team Development X

Commit Update | History | Activity | Versions |

Comment:
 Category and Attraction Business Component in order to insert a new category ("Tourist Site") and change Beijing "monument" attractions to the new category Recent Comments...

Pending Commits (7/7) Ignored Objects Y ↕ ↻

<input checked="" type="checkbox"/>	<input type="checkbox"/>	Name	Type	Description	Modified On	Module	Local State	Last Synchronized	User
<input checked="" type="checkbox"/>		Attraction	Transaction	Attraction	10/7/2020 1...	Root Module	Modified	7/7/2020 16:36	ARTECH(CFerna...
<input checked="" type="checkbox"/>		AttractionWith...	Transaction	Attraction Without Para...	7/7/2020 17...	Root Module	Inserted	3/7/2020 15:34	ARTECH(CFerna...
<input checked="" type="checkbox"/>		CategoriesAn...	Web Panel	Categories And Attractio...	13/7/2020 1...	Root Module	Inserted	3/7/2020 15:34	ARTECH(CFerna...
<input checked="" type="checkbox"/>		Category	Transaction	Category	13/7/2020 1...	Root Module	Modified	3/7/2020 15:34	ARTECH(CFerna...
<input checked="" type="checkbox"/>		InsertCategor...	Procedure	Insert Category Update...	13/7/2020 1...	Root Module	Inserted	3/7/2020 15:34	ARTECH(CFerna...
<input checked="" type="checkbox"/>		MassiveInsert...	Web Panel	Massive Insert Remove	13/7/2020 1...	Root Module	Inserted	3/7/2020 15:34	ARTECH(CFerna...
<input checked="" type="checkbox"/>		WorkWithAttr...	Work With for Web	Work With Attraction	7/7/2020 16...	Root Module	Modified	3/7/2020 15:34	ARTECH(CFerna...

Remind me to move changes to... Commit

Por último, vamos enviar as alterações realizadas para o servidor.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications