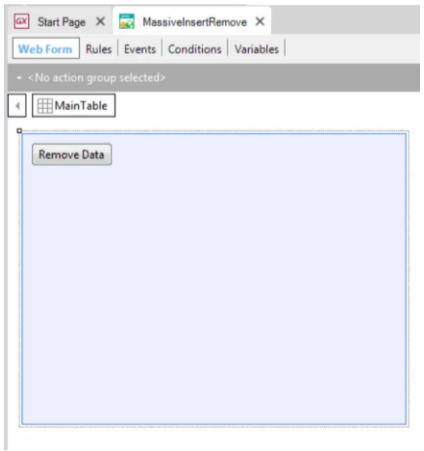
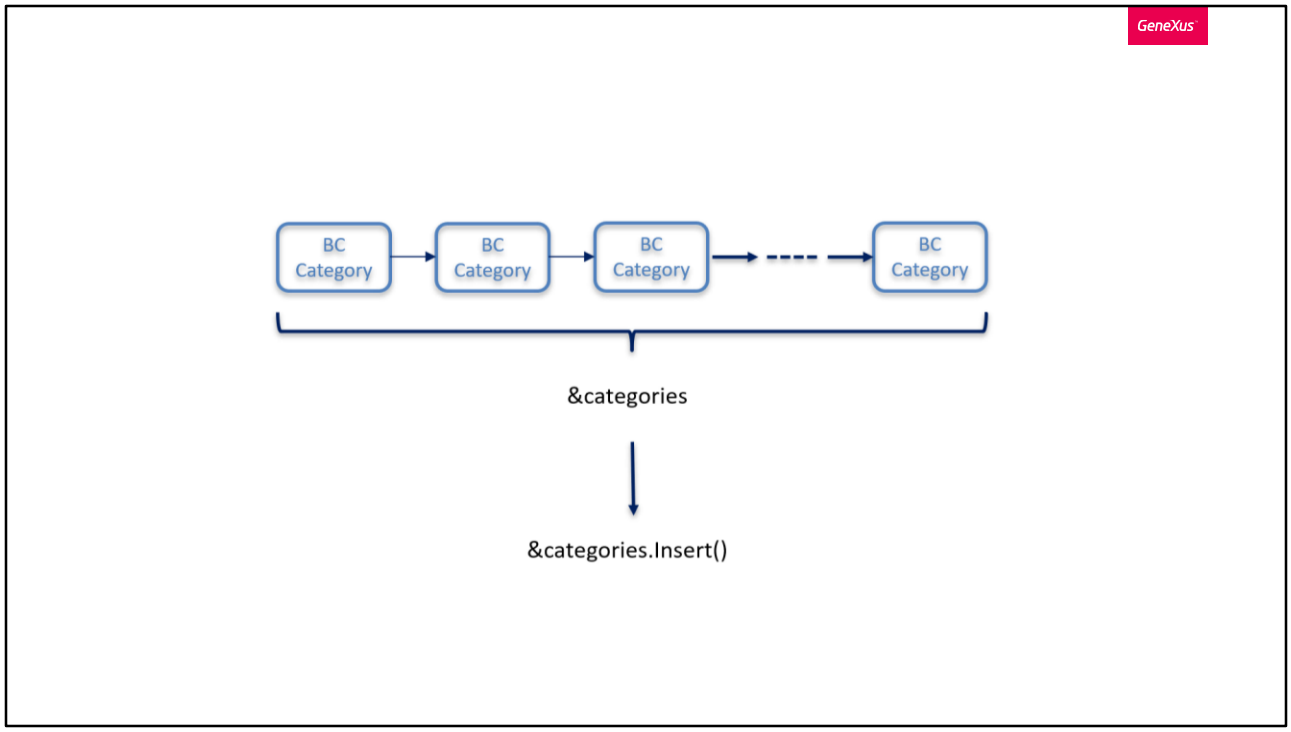
Populando com dados usando Business Component e Data Provider

GeneXus



Suponhamos que as tabelas de países, cidades, etc. já têm dados. Como anteriormente eliminamos os dados das atrações e categorias, nosso objetivo será inicializar as tabelas de categorias e atrações, com dados, para começar a partir de tabelas não vazias.

Para fazer isso, abrimos o web panel "MassiveInsertRemove", criado no vídeo anterior, adicionamos um botão “Initialize data” e usaremos o data provider em conjunto com os business components estudados recentemente.

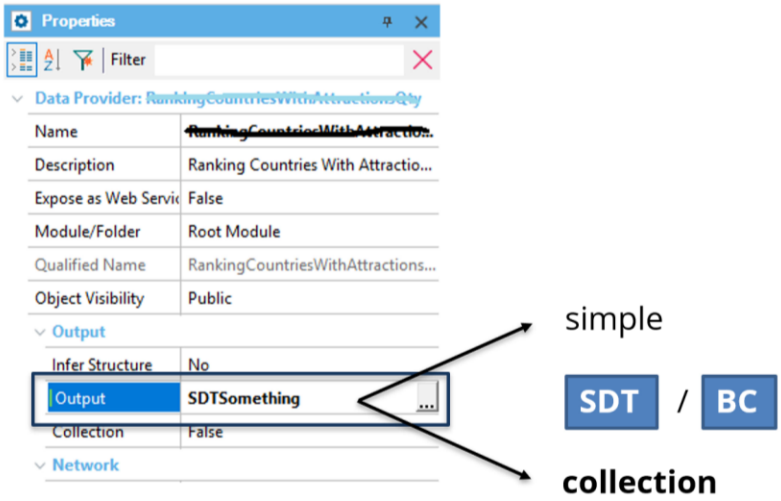


Se obtivermos uma variável coleção de itens do tipo business component de Category, carregada com as categorias que queremos inserir no banco de dados, poderemos aplicar o método Insert() a essa variável coleção, pois como mencionamos no vídeo anterior, isso permite fazer o Insert de todos os itens, ou seja, de todos os business components da coleção.

Nosso problema é reduzido, então, a obter essa coleção. Como vamos fazer?

GeneXus

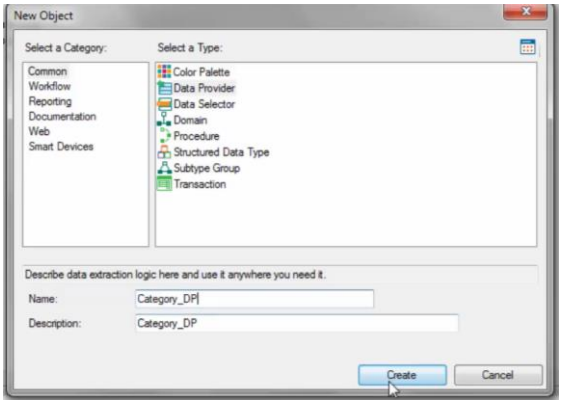
Data Provider



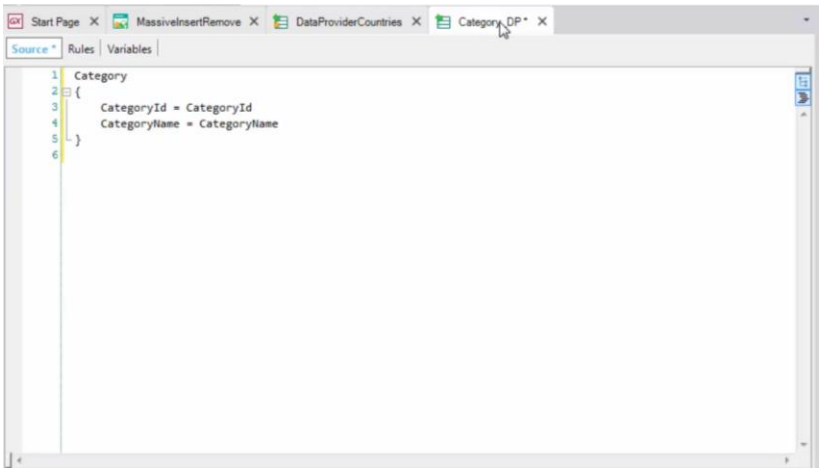
Até agora sabíamos que um Data Provider nos permitia retornar dados estruturados, tanto simples como coleção.

No nosso caso, queremos retornar uma coleção de categorias, mas estas categorias não são tipos de dados estruturados, mas **business components**.

No entanto, um business component é exatamente como um SDT no que se refere à sua estrutura. Assim, os data providers também nos permitirão carregar e devolver business components, tanto simples como coleções.



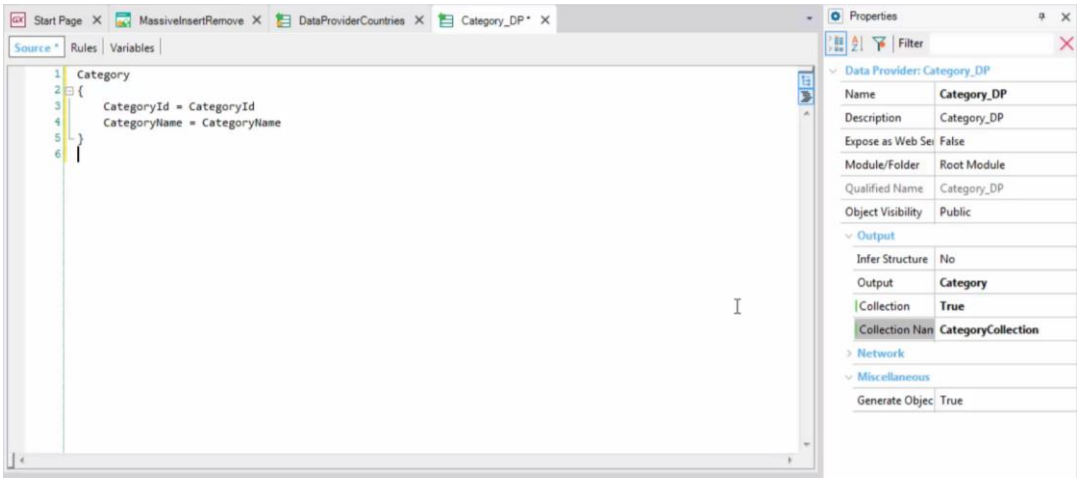
É daí que virá a nossa solução. Criamos um data provider para carregar as categorias. Vamos chamá-lo de Category_DP.



Arrastamos a transação Category dentro do Source do data provider e vemos que nos escreve a estrutura da transação. Observemos que à esquerda temos os elementos do business component, que estarão em memória, e que têm o mesmo nome que os atributos, mas não são.

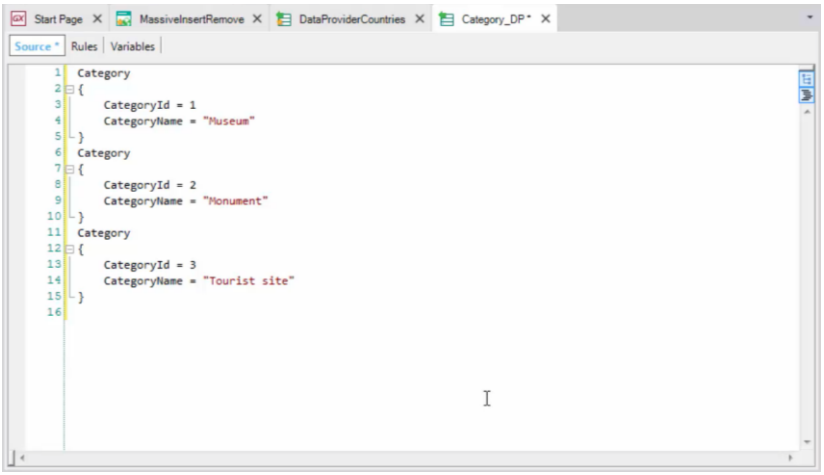
Enquanto que à direita, por padrão, estão sendo mostrados, desta vez sim, os atributos da tabela correspondente, a partir de onde o data provider obterá os dados para carregar o BC que está em memória.

Se isso é o que gostaríamos, o data provider deve retornar uma coleção deste business component, uma vez que a tabela tem muitos registros.



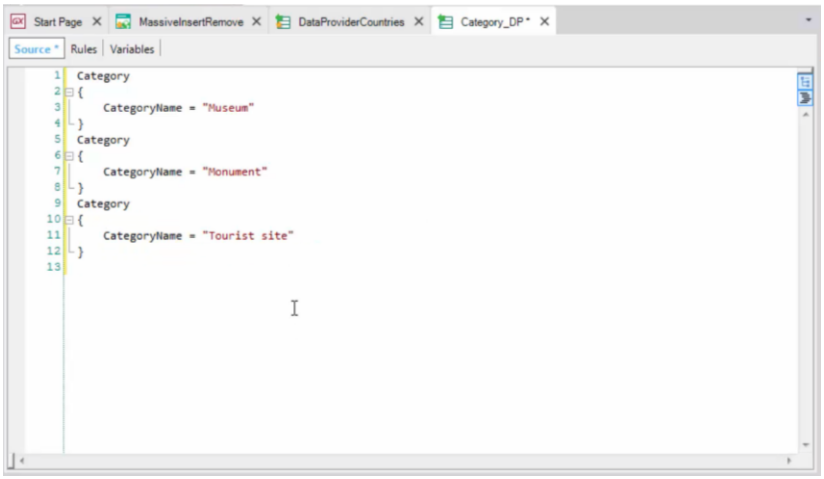
Se vamos às propriedades, vemos que a propriedade Output assumiu o valor do business component, mas a propriedade collection não está como True, como precisamos.

Então, a alteramos e nos aparece nova propriedade de Collection name, que por padrão assume o nome do data provider. O alteramos para CategoryCollection.

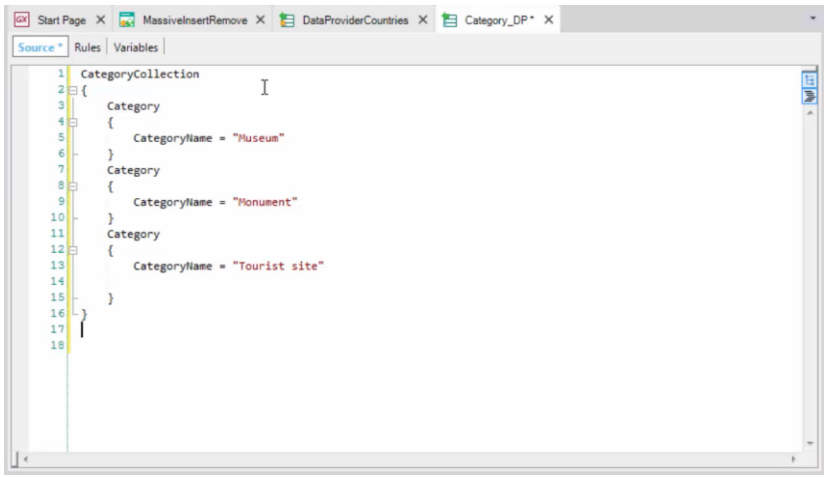


E também, não queremos carregar essa coleção com dados do banco de dados, mas queremos atribuir-lhes novos valores, especificados por nós.

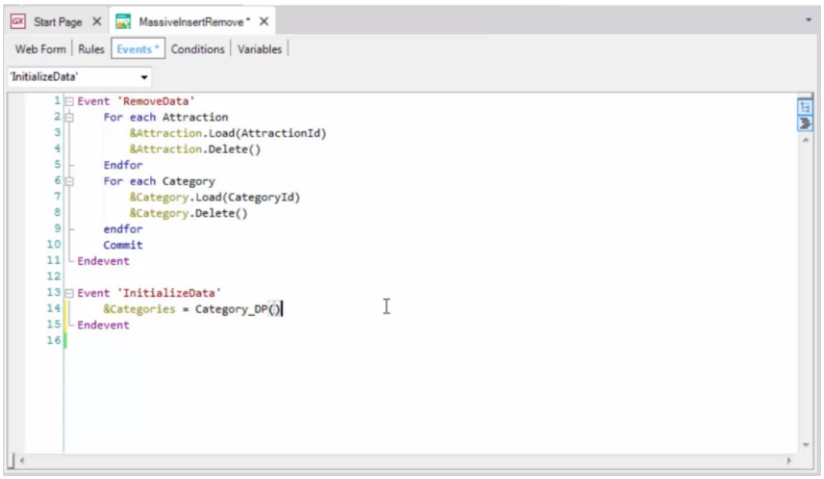
Portanto, um por um, vamos escrever grupos associados aos itens da coleção:



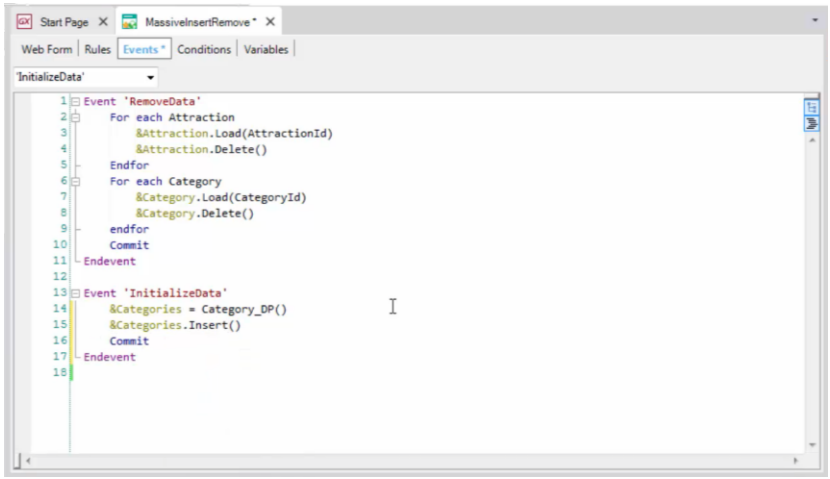
Como CategoryId é um atributo autonumerado, não precisamos atribuir-lhe valor quando queremos inserir um registro, que será o que faremos mais tarde, então diretamente excluímos essa atribuição:



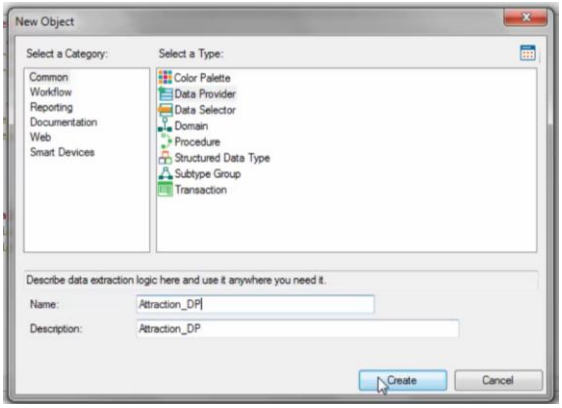
E como o que queremos retornar é uma coleção de nome CategoryCollection, mesmo que não seja necessário--porque ao colocar para a propriedade Collection o valor True, o data provider já sabe que retornará uma coleção, à qual chamará dessa maneira--, para esclarecer o código podemos explicitar o que GeneXus já interpreta: encerrando todos os grupos Category dentro do grupo CategoryCollection, que corresponde à coleção.



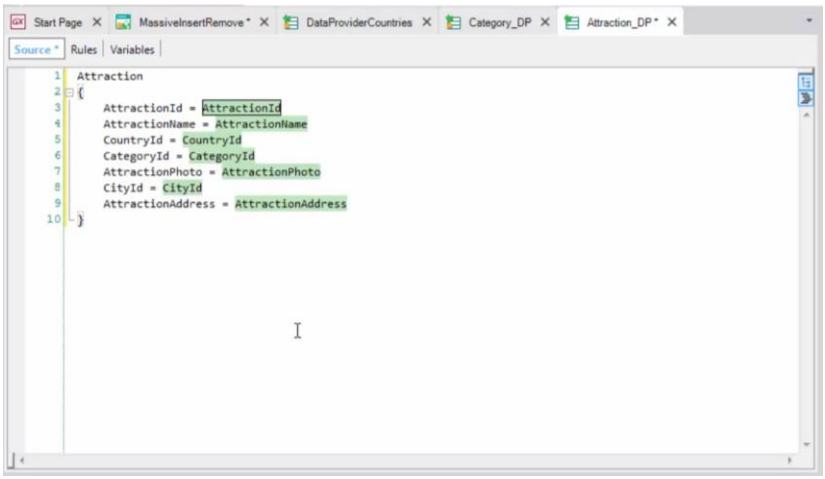
Agora só precisamos chamar esse Data Provider a partir do evento associado com o botão do web panel:



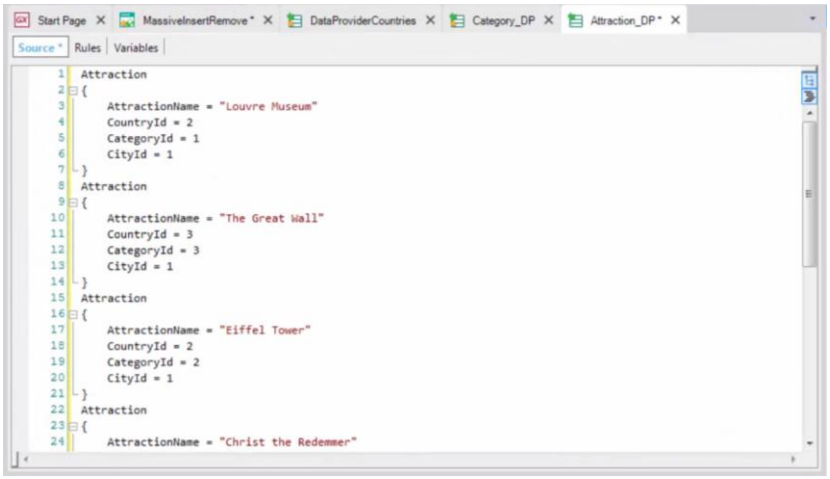
E em seguida inserir no banco de dados:



Agora teríamos que inicializar a tabela de atrações. Da mesma forma, criaremos um data provider, Attraction_DP.

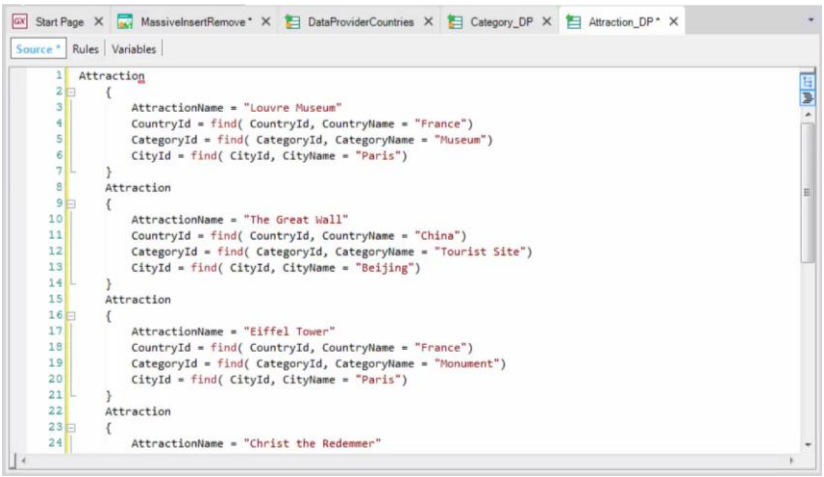


Arrastaremos a Transação (da qual já obtivemos o business component) e vemos que cada elemento do business component é inicializado por padrão com o atributo correspondente da tabela.



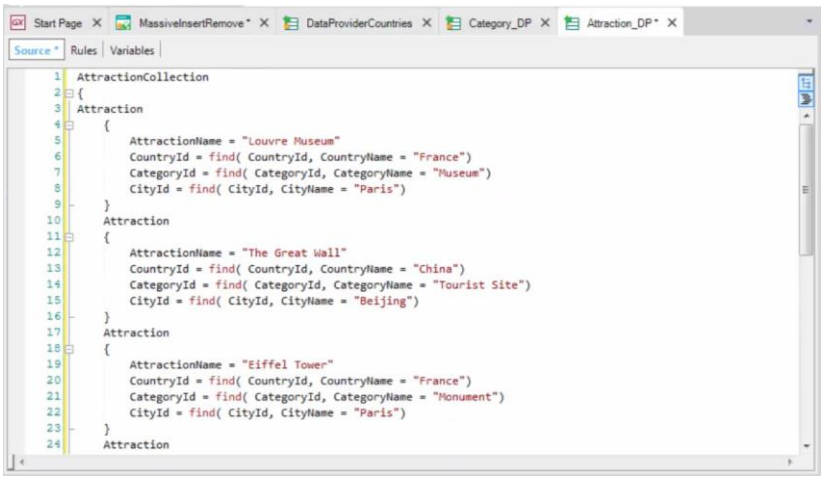
Novamente, vemos que somente os atributos presentes fisicamente na tabela são levados em conta, não aqueles que são inferidos na transação ou fórmulas.

Como não nos interessa carregar atrações existentes (pois na verdade, executamos esse data provider para carregar os primeiros dados), eliminamos todos estes atributos, e incluiremos manualmente estes valores. Além disso, como o Id é autonumerado, não é necessário atribuir valor para este elemento do business component. As fotos das atrações serão atribuídas mais tarde, por isso também removemos este atributo.



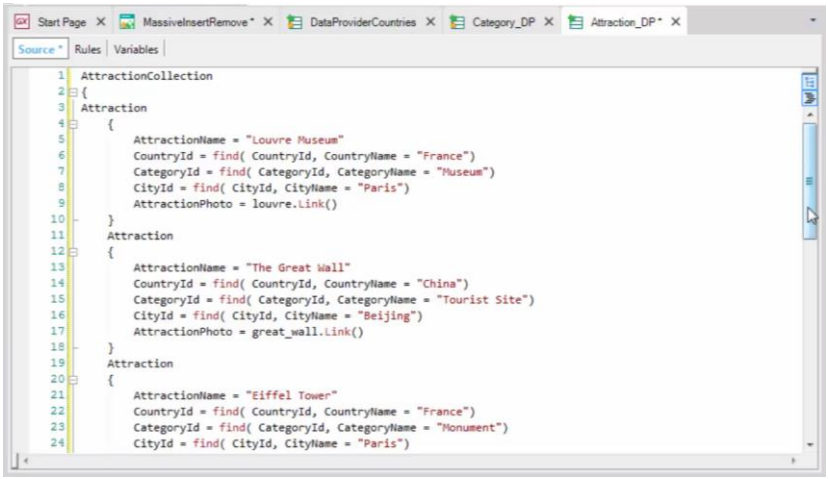
Como aqui estamos atribuindo os valores de CountryId, CityId e CategoryID de memória, eles podem não existir nas tabelas correspondentes. Se algum dos valores não existirem, no momento da tentativa de inserir os registros com o business component, os controles de integridade referencial correspondentes serão disparados e a inserção falhará.

Para evitar a atribuição de valores que podem não existir, usaremos a fórmula Find para localizar os identificadores corretos a partir dos nomes do país, da cidade ou categoria.



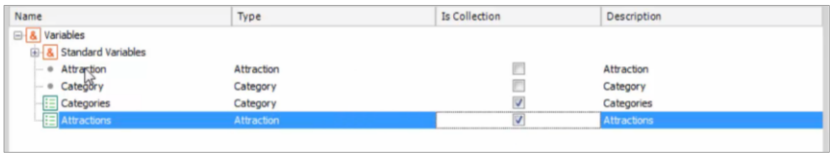
Observemos que as fórmulas Find estão acessando o banco de dados apenas para procurar os identificadores correspondentes aos nomes que usamos, mas o restante dos valores atribuídos ao business component são fixos.

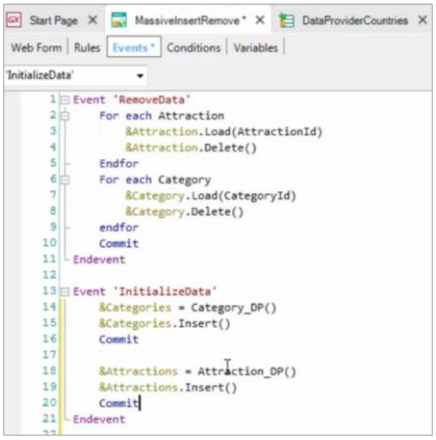
Como fizemos com o Data Provider das categorias, devemos colocar a propriedade Collection em True, já que vamos retornar muitas atrações e também ajustaremos a notação no source fechando os grupos dentro do grupo AttractionCollection, para indicar que é uma coleção de atrações.



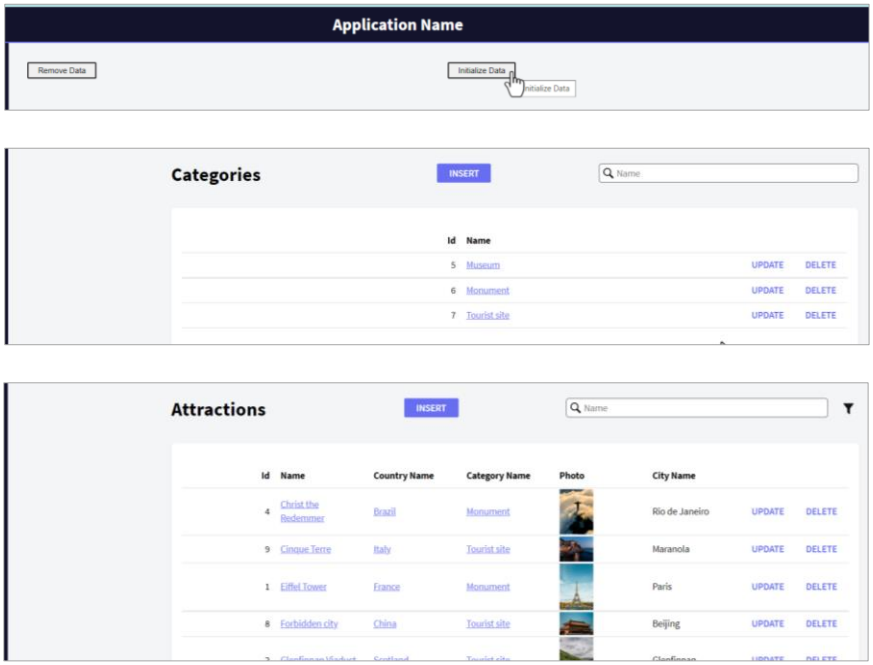
Para carregar também as fotos das atrações, uma possibilidade é inseri-las primeiro como objetos de imagem na KB...

E, em seguida, para cada grupo do Data Provider, basta atribuir a AttractionPhoto o nome da imagem ponto link.

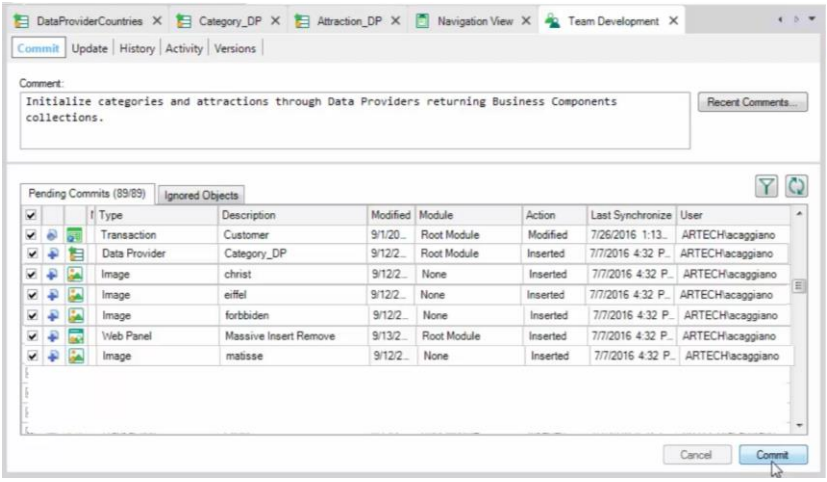




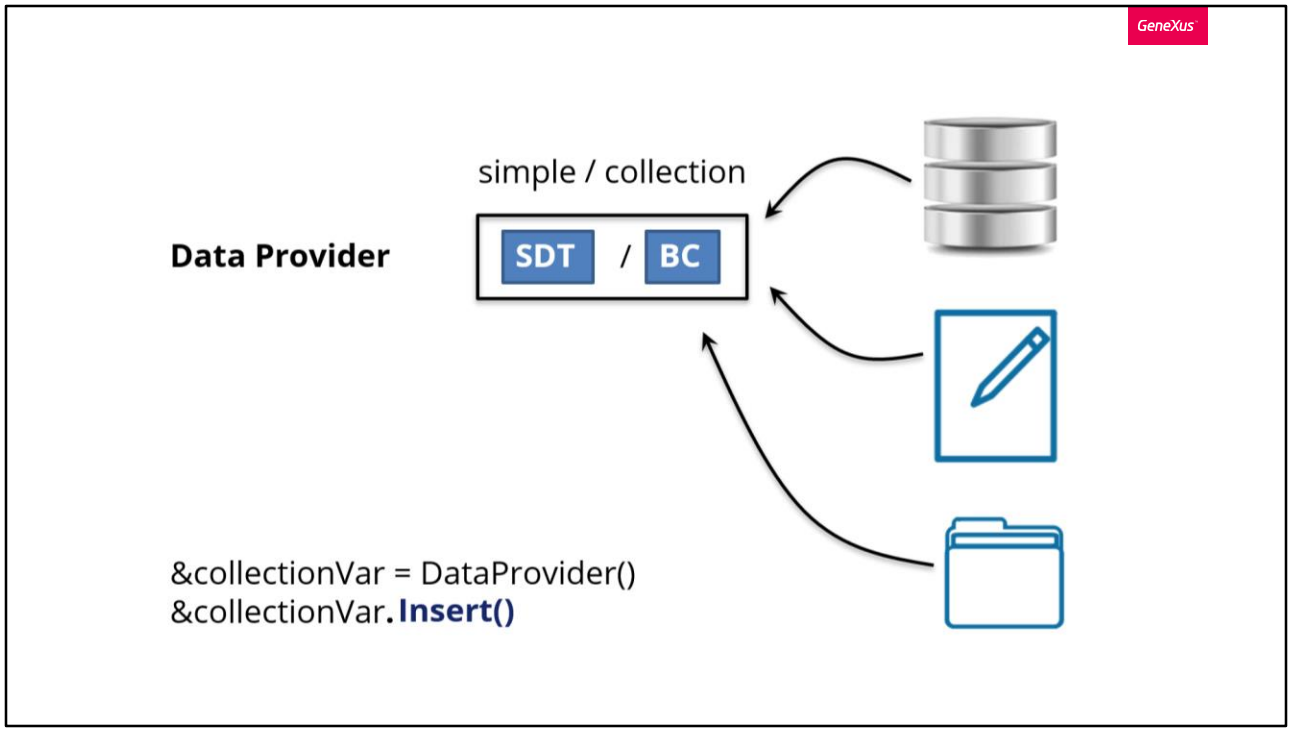
Agora falta somente chamar o Data Provider para retornar a coleção carregada....



Notemos que para ser capaz de inserir as atrações, primeiro temos que ter criadas as categorias, por isso a ordem é aquela que usamos no código do evento.



Vamos fazer um commit no GeneXus Server.



Neste vídeo vimos como um data provider não só permite carregar uma estrutura com dados do banco de dados... mas também a partir de dados fixos... mas também pode fazê-lo através de outras fontes externas, como poderá estudar em cursos mais avançados.

Também vimos que um data provider permite carregar a estrutura de um business component (e não apenas de um SDT), tanto simples como coleção.

Por último, vimos que no caso da estrutura ser do tipo coleção, poderemos aplicar métodos que afetam todos os itens da coleção, em uma única operação, como por exemplo, insert() e delete().

GeneXus™

training.genexus.com
wiki.genexus.com