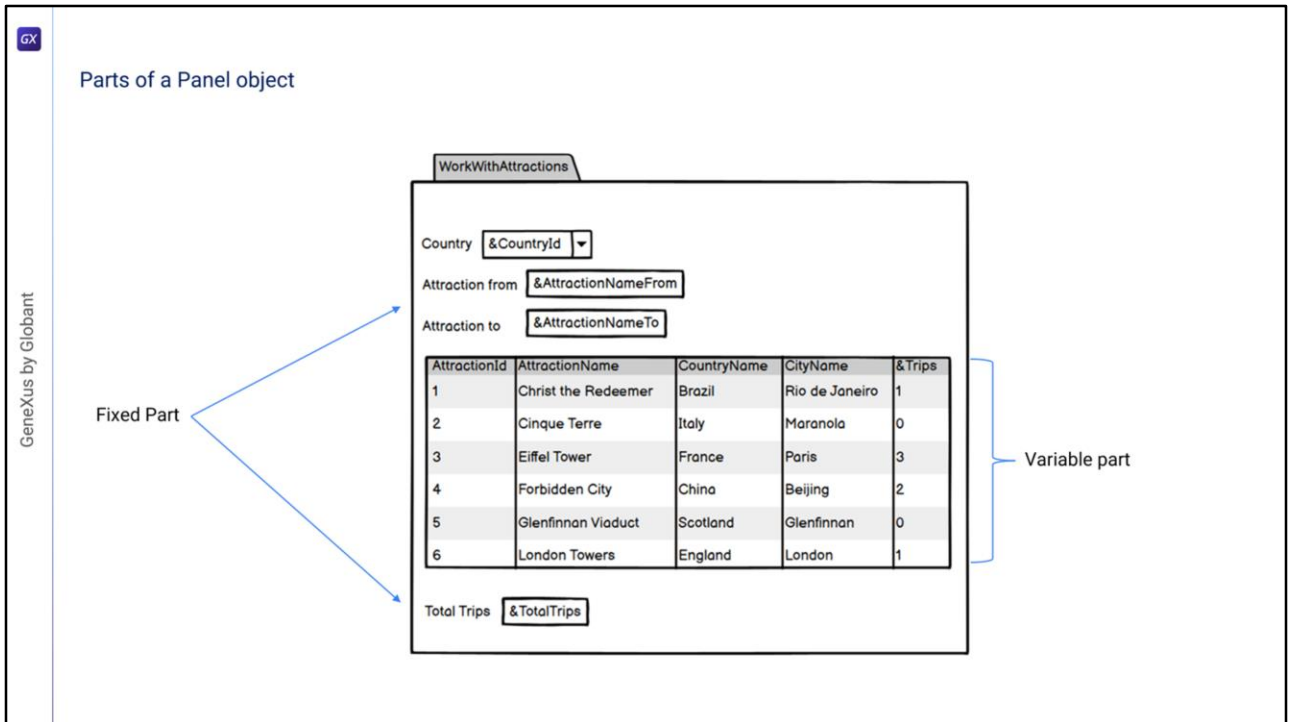


# Data loading logic and base tables in a Panel



Vanesa Fernández

Neste vídeo estudaremos como é carregada a tela de um objeto Panel dependendo dos componentes da interface gráfica presentes nela, como são determinadas suas tabelas base e como fazemos para pesquisar, ordenar e filtrar a informação em tela.



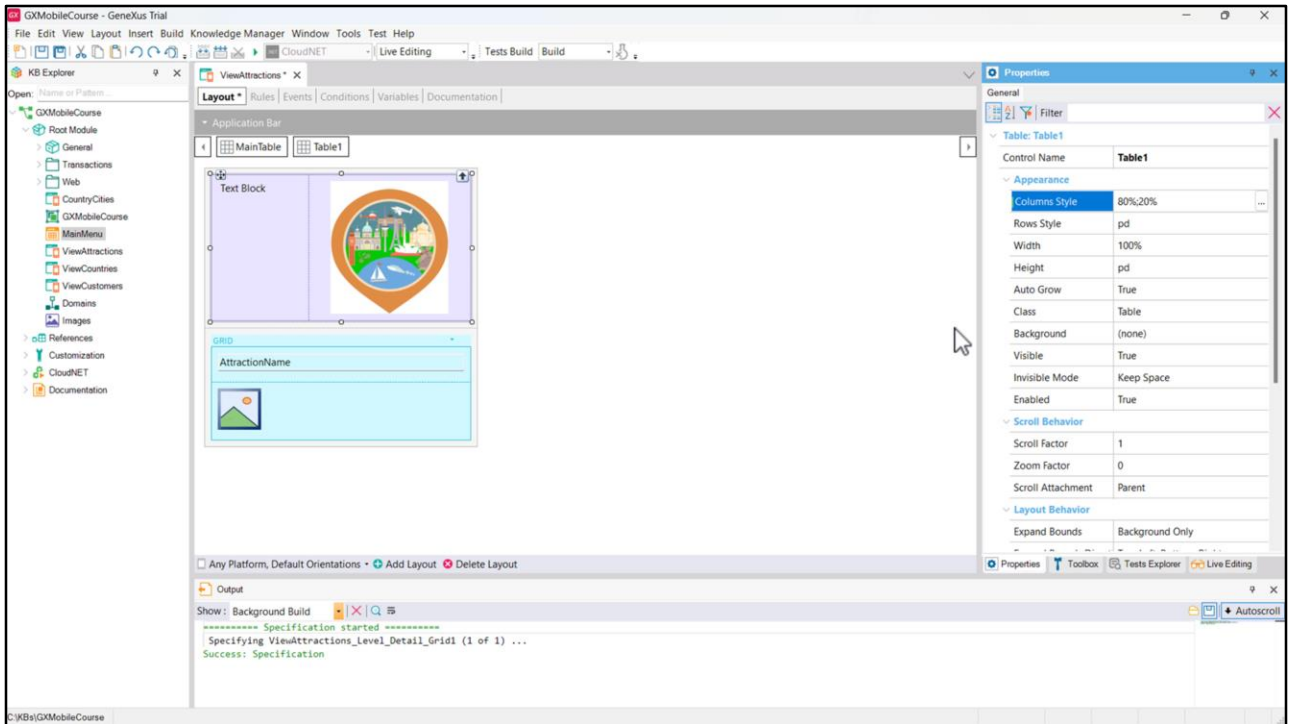
Em um objeto Panel podemos identificar duas partes distintas:

Chamamos de parte fixa ou plana a parte que contém tudo aquilo que está no form e que não está incluído em nenhum grid. Isto inclui controles como textblocks, combos, botões e outros que podemos arrastar a partir da barra de ferramentas.

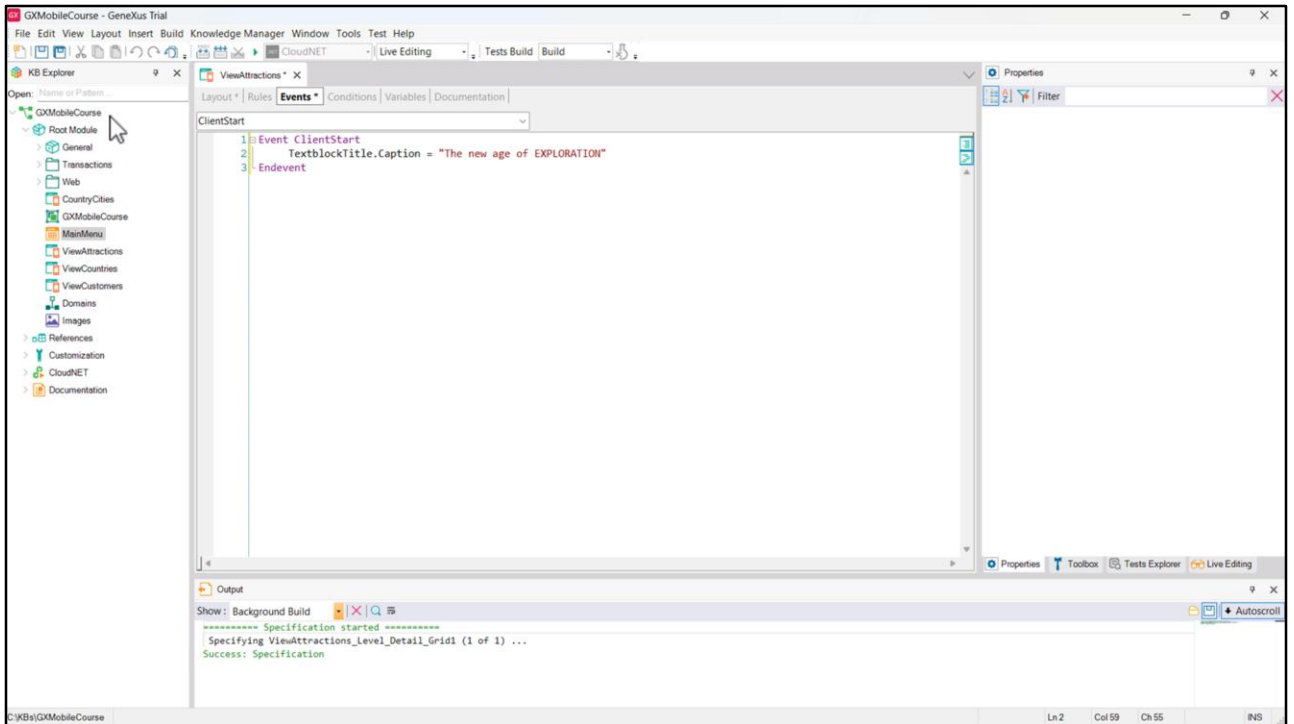
A segunda parte chamaremos de variável, e é composta pelo grid ou grids que foram incluídos no form.

Em um objeto Panel sempre teremos uma parte fixa, e podemos ter uma parte variável para cada um dos grids que tenha o Panel.

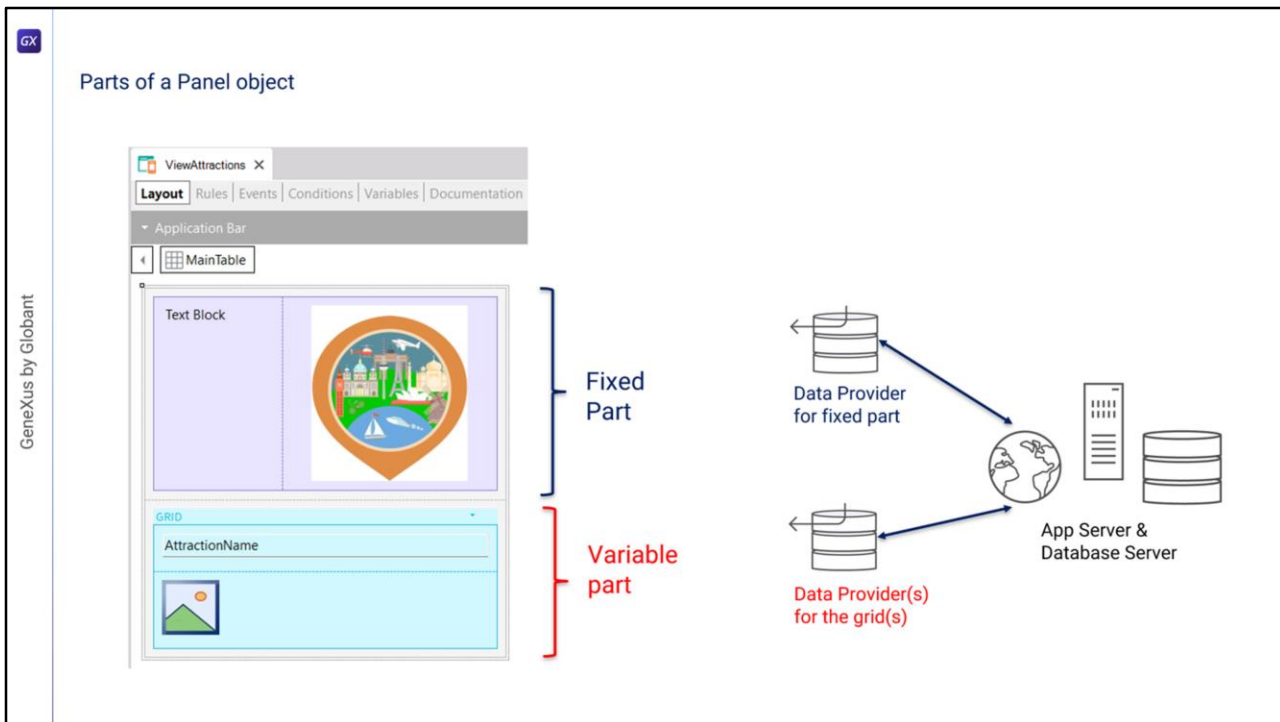
Os objetos Panel se diferenciam dos WebPanels porque a parte fixa é carregada de forma independente da parte variável.



Antes de continuar, vamos ao GeneXus e adicionamos alguns controles à parte fixa do Panel ViewAttractions que criamos. Acima e fora do grid, adicionamos uma tabela com um Textblock chamado TextblockTitle, alinhamento horizontal centralizado e alinhamento vertical médio, e uma imagem (a mesma que estamos usando no menu, já que não temos interesse aqui em entrar no design da tela), com alinhamento horizontal centralizado. Configuramos a propriedade Columns Style da tabela para que a primeira coluna -que contém o Textblock- ocupe 80% do espaço total, e a segunda coluna -que contém a imagem- os 20% restantes.

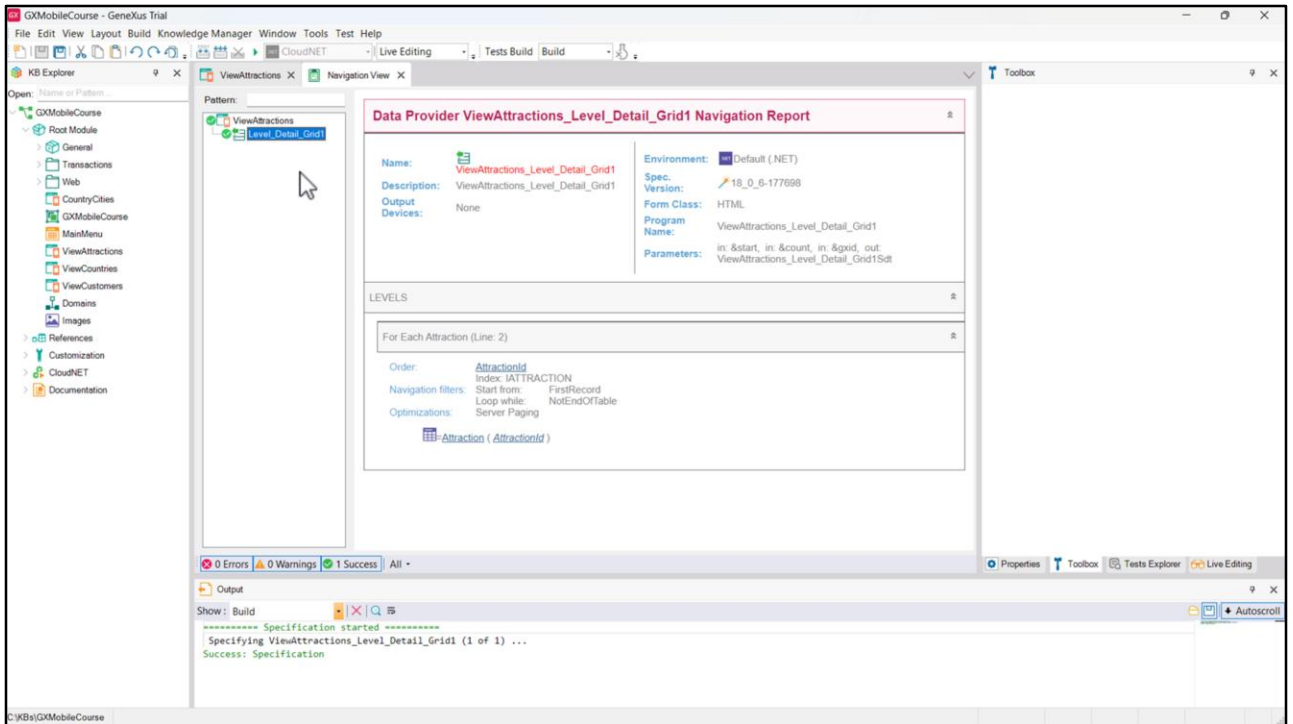


No evento ClientStart atribuímos à sua propriedade Caption o título que queremos ver em tela, "The new age of EXPLORATION", salvamos e continuamos com a análise da carga dos dados do Panel.



Para cada parte (fixa e grid), GeneXus irá gerar automaticamente Data Providers independentes que serão publicados como serviços no servidor e serão os que acessarão a base de dados para obter a informação necessária para carregar a parte fixa e a parte variável do Panel.

Estes Data Providers não veremos na Base de Conhecimento já que GeneXus se encarregará de gerá-los e mantê-los, mas poderemos ver quais dados eles acessam se visualizarmos sua lista de navegação.



Na lista de navegação do Panel ViewAttractions vemos que sob o nó correspondente ao Panel aparece uma entrada denominada Level\_Detail\_Grid1, que corresponde à sua parte variável, que neste caso está composta pelo Grid1.

Se o Panel tivesse mais de um grid, apareceria um nó de detalhe para cada grid, pois cada um terá uma navegação própria.

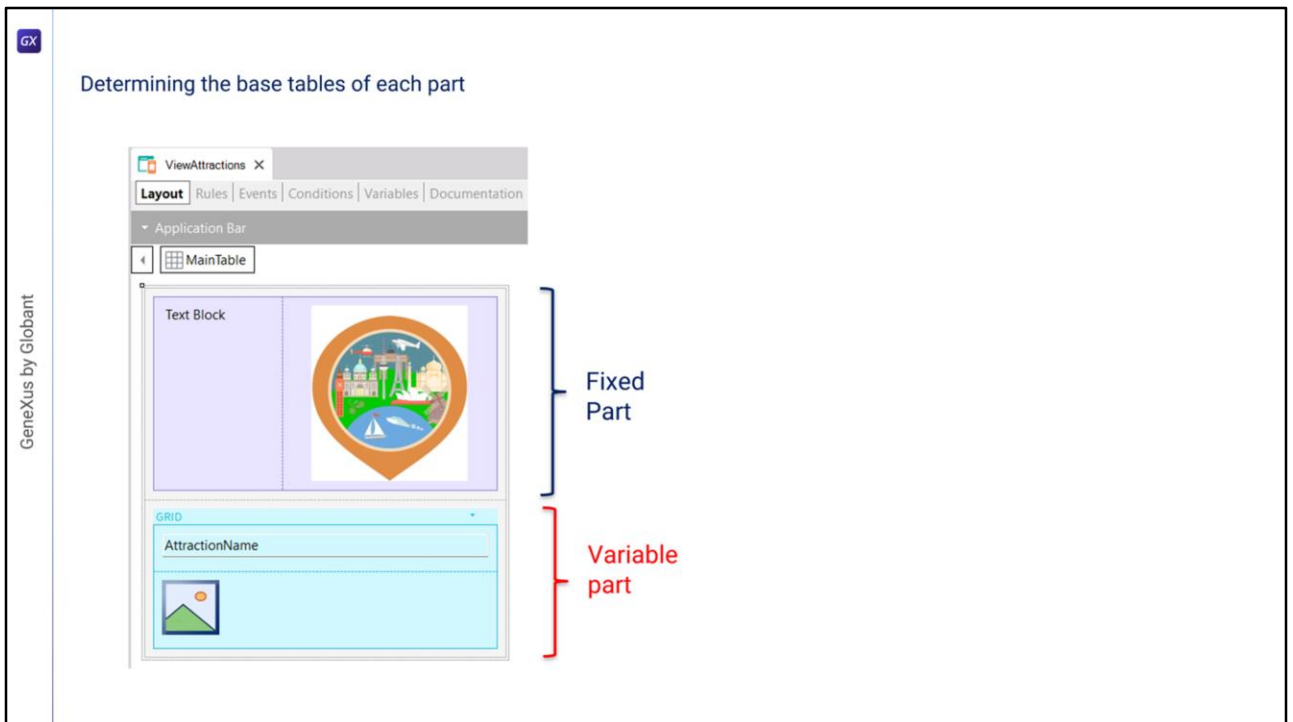
Então, selecionamos o nó Level\_Detail\_Grid1 e vemos que recupera os dados do Data Provider ViewAttractions\_Level\_Detail\_Grid1.

Este é o Data Provider que foi criado automaticamente para carregar o Grid1, publicado como serviço no servidor e invocado pelo Panel para acessar a base de dados e recuperar as atrações.

Se olharmos onde diz Environment, observamos que o Data Provider foi gerado em .NET como parte do código do backend.

Na seção de parâmetros, vemos que o Data Provider recebe alguns dados do grid e devolve carregado um SDT chamado View\_Attractions\_Level\_Detail\_Grid1, que conterà os dados para carregar o grid.

A informação da lista é similar à que veríamos com um Web Panel com tabela base Attraction, já que o grid do Panel está percorrendo a tabela Attraction para mostrar as atrações turísticas.



Como a parte fixa e as partes variáveis do Panel são carregadas separadamente, as tabelas base de cada parte são completamente independentes uma da outra. Saber como GeneXus determina quais tabelas deve acessar e a forma como as percorre é vital para que nossa aplicação funcione corretamente.

No exemplo vemos elementos na parte fixa do Panel e um grid, portanto serão determinadas as tabelas base da parte fixa e da parte variável separadamente. Pode acontecer que a parte fixa tenha tabela base e o grid não, que o grid tenha tabela base e a parte fixa não, que ambas as partes tenham tabela base ou que nenhuma parte tenha.

É importante lembrar que nisto os Panels funcionam de forma diferente dos WebPanels, pois em um WebPanel com grid único, se houver tabela base, a tabela base do WebPanel é única e não existem duas tabelas base diferentes separadas para parte fixa e grid, como acontece com os Panels.

GeneXus by Globant

GX

### Determining the base tables of each part

The screenshot shows the GeneXus IDE interface for a panel named 'ViewAttractions'. The 'Layout' tab is active, showing a design with two main sections. The top section, labeled 'Fixed Part' with a blue bracket, contains a 'Text Block' and a circular image of a cityscape. The bottom section, labeled 'Variable part' with a red bracket, contains a 'GRID' with a single column labeled 'AttractionName' and a small landscape icon below it. The 'Application Bar' at the top shows 'MainTable' selected.

#### Attributes involved in determining the **Fixed Part base table**:

- Attribs. in fixed part of panel (form)
- Attribs. outside For Each commands in Refresh event and events of buttons or controls in fixed part and Application Bar
- Attribs. in Conditions Tab

Dado então que em um objeto Panel a parte fixa e o grid determinam navegações independentes e cada parte terá sua tabela base, é como se houvesse dois for eachs paralelos.

Para determinar a tabela base da parte fixa, serão levados em consideração os atributos que pertencem à parte fixa do form e os atributos que pertencem aos eventos associados à parte fixa, desde que esses atributos estejam fora de um comando For Each.

Estes eventos são o evento Refresh e os eventos associados a botões ou controles da parte fixa, incluindo aqueles da Application Bar.

Além disso, para determinar a tabela base da parte fixa, também devem ser considerados os atributos da Tab Conditions do objeto Panel.



GeneXus by Globant

ViewAttractions x

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

Text Block

GRID

AttractionName

Fixed Part

Variable part

Attributes involved in determining the **Fixed Part base table**:

- Attribs. in fixed part of panel (form)
- Attribs. outside For Each commands in Refresh event and events of buttons or controls in fixed part and Application Bar
- Attribs. in Conditions Tab

Attributes involved in determining the **Variable Part (grid) base table**:

- Attribs. in grid columns
- Attribs. in Order, Search, Advanced Search and Conditions
- Attribs. outside For Each in Load event and events of buttons or controls inside the grid
- Attribs. in Conditions Tab

Grid Base Trn property assigned

Para determinar a tabela base da parte variável, neste caso do grid, serão levados em consideração os atributos incluídos nas colunas do grid, tanto visíveis quanto não visíveis, os atributos referenciados na Order, Search, Advanced Search e Conditions do grid, os atributos que pertencem ao código do evento Load desde que estejam fora de cláusulas For Each e presentes nos eventos de botões ou controles dentro do grid.

Também serão levados em consideração os atributos que estão na tab Conditions, que serão considerados para a determinação da tabela base de todos os grids que estão incluídos no Panel.

Por último, o grid também terá tabela base se sua propriedade Base Trn tiver sido atribuída com uma transação base. Neste caso, os atributos que estão nas outras partes devem pertencer à tabela estendida da tabela associada à transação base.

GeneXus by Globant

### Determining the base tables of each part

The image shows two panels from the GeneXus IDE. The left panel displays a form design for 'ViewAttractions'. It is divided into two sections: a 'Fixed Part' (top) and a 'Variable part' (bottom). The fixed part contains a 'Text Block' with an image and a 'MainTable'. The variable part contains a 'GRID' with an 'AttractionName' field. The right panel shows the 'Conditions' tab of the code editor, displaying the following code:

```

1
2
3
ClientStart
1 Event ClientStart
2   TextblockTitle.Caption = "The new age of EXPLORATION"
3 Endevent

```

Fixed Part: NO BASE TABLE

No exemplo que vemos, a parte fixa é composta apenas por uma tabela, um texto e uma imagem no form, ou seja, não há atributos. Também não há atributos na tab Conditions do Panel. Não há evento Refresh nem eventos de controles do form ou botões na ApplicationBar, ou seja, a parte fixa deste Panel não tem tabela base.

GeneXus by Globant

### Determining the base tables of each part

Layout | Rules | Events | **Conditions** | Variables | Documentation |

1

Layout | Rules | **Events** | Conditions | Variables | Documentation |

ClientStart

```

1 Event ClientStart
2   TextblockTitle.Caption = "The new age of EXPLORATION"
3 EndEvent

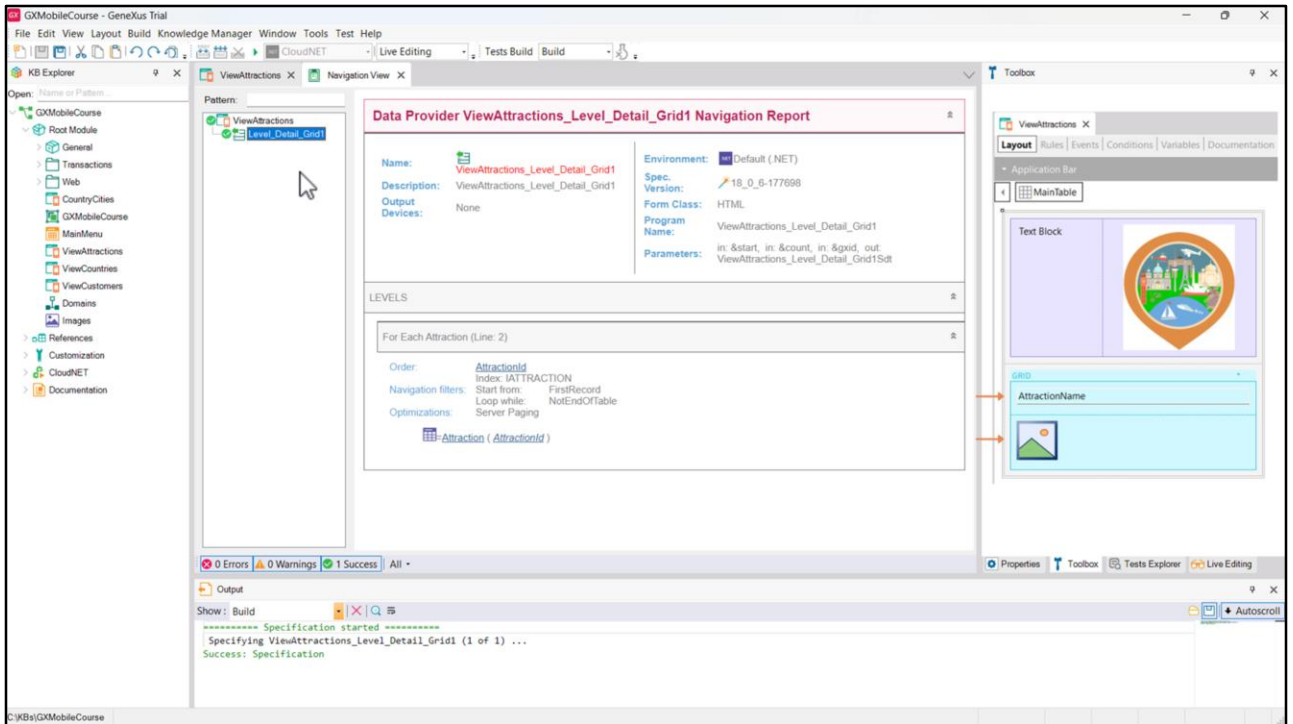
```

**Fixed Part: NO BASE TABLE**

Attributes in the grid: AttractionName, AttractionPhoto

**Variable part base table: ATTRACTION**

Se analisarmos a parte variável, vemos que existem atributos nas colunas do grid, que são: AttractionName e AttractionPhoto. Não há atributos nas propriedades Order, Search, nem Conditions do grid e também não está atribuído o valor da propriedade Base Trn, não há atributos no evento Load nem em eventos de controles dentro do grid. Portanto, a parte variável do Panel, composta pelo grid, terá tabela base Attraction. Isto significa que GeneXus construirá um Data Provider como serviço no backend e recuperará os dados que carregarão o grid.



Vejamos a lista de navegação do Panel para constatar qual é a tabela base de sua parte variável: diz For Each Attraction, então a tabela base é efetivamente ATTRACTION, como havíamos deduzido e mencionado previamente.

Mais abaixo vemos que acessa a tabela Attraction para recuperar os dados de AttractionName e AttractionPhoto.

Em resumo, vemos que pelo simples fato de que o grid tenha atributos, neste caso em suas colunas, GeneXus foi capaz de determinar automaticamente qual tabela devia percorrer e criar tudo o que fosse necessário para recuperar os dados necessários a partir da base de dados.

Another example of base table determination

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

Customer Id CustomerId

Customer Name CustomerName

GRID

| AttractionId | AttractionName | AttractionDescription |
|--------------|----------------|-----------------------|
|              |                |                       |
|              |                |                       |
|              |                |                       |

Trip

- TripId
- TripDate
- TripDescription
- CustomerId
- CustomerName
- CustomerLastName
- CustomerFullName
- Attraction
- AttractionId
- AttractionName
- CountryId
- CountryName
- CityId
- CityName
- TripAttractionDuration

Customer

- CustomerId
- CustomerName
- CustomerAddress
- CustomerPhone
- CustomerEmail
- CustomerPhoto
- CountryId
- CountryName

Attraction

- AttractionId
- AttractionName
- AttractionDescription
- AttractionPhoto
- CountryId
- CountryName
- CityId
- CityName

Country

- CountryId
- CountryName
- CountryFlag
- City
- CityId
- CityName

ViewCustomers

Layout Rules Events Conditions Variables Documentation

Events

- Event "Trips"
- AttractionsVisitedByCustomer(CustomerId)
- EndEvent

Vejam os outros exemplos de determinação de tabelas base. A agência de viagens deseja ter um Panel onde para um determinado cliente recebido por parâmetro sejam mostradas as atrações visitadas por esse cliente.

Esta informação está modelada na transação Trip onde cada viagem tem um cliente (observemos que CustomerId é chave estrangeira e o atributo CustomerName é inferido). Cada viagem também possui muitas atrações que são visitadas, representadas pelo segundo nível da transação Trip. Também são mostradas, a transação Customer que tem os dados dos clientes, a transação Attraction com os dados das atrações e a transação Country com os dados de países e cidades.

Para testar este exemplo, invoque este Panel a partir de outro objeto de sua KB, passando como parâmetro o identificador do cliente.

GeneXus by Globant

### Determining the base table of a fixed part

The screenshot displays the GeneXus IDE interface for a fixed part. The main table, 'MainTable', contains the following fields:

- Customer Id (CustomerId)
- Customer Name (CustomerName)
- GRID (AttractionId, AttractionName, AttractionDescription)

The 'Rules' tab shows a parameter definition: `1 Parm(in: &CustomerIdentifier);`

The 'Conditions' tab shows a filter rule: `1 CustomerId = &CustomerIdentifier;`

The 'Events' tab is currently empty.

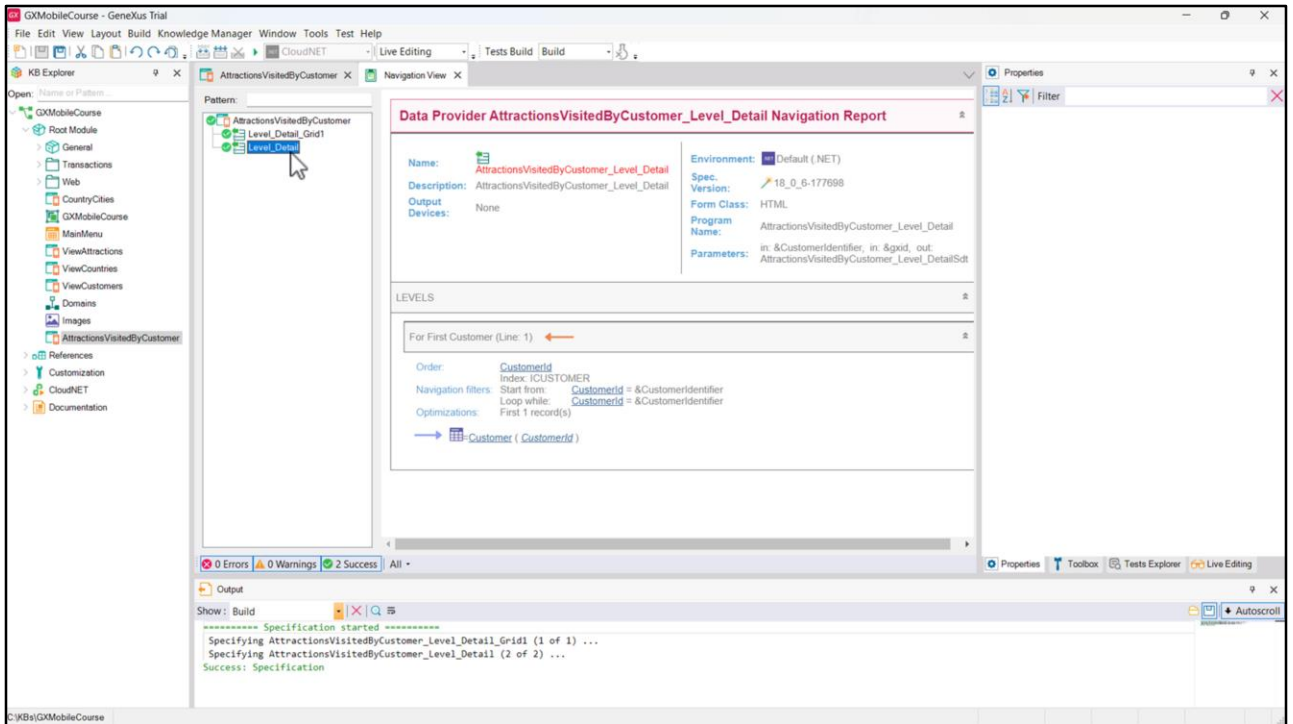
The 'Customer Structure' tree shows the following fields:

- CustomerId
- CustomerName
- CustomerAddress
- CustomerPhone
- CustomerEmail
- CustomerPhoto
- CountryId

**Fixed part base table: CUSTOMER**

Vamos analisar primeiro a parte fixa do Panel. Além dos atributos presentes no form, se procurarmos atributos em outras partes do Panel, vemos que nas regras existe somente a variável `&CustomerIdentifier` que recebe o identificador do cliente a ser mostrado e na tab Conditions encontramos um filtro que garante que serão exibidos os dados somente do cliente recebido por parâmetro. Não temos nada na aba eventos.

Significa que os únicos atributos que devemos analisar para a parte fixa são `CustomerId` (presente no form e na tab Conditions) e `CustomerName` presente no form. A tabela base é então `CUSTOMER`, pois contém ambos os atributos.



Se observarmos a lista de navegação do nó Level\_Detail correspondente à parte fixa, vemos que a tabela base é Customer, como havíamos determinado anteriormente e que é acessada a referida tabela para recuperar os dados dos atributos CustomerId e CustomerName.

GeneXus by Globant

### Determining the base table of a variable part (grid)

The screenshot displays the GeneXus IDE interface. On the left, the 'MainTable' is shown with columns 'Customer Id' and 'CustomerName'. Below it, a 'GRID' is defined with columns 'AttractionId', 'AttractionName', and 'AttractionDescription'. On the right, the 'Rules' tab shows a parameter declaration: `1 Parm(in: &CustomerIdentifier);`. The 'Conditions' tab shows a condition: `1 CustomerId = &CustomerIdentifier;`. The 'Events' tab is currently empty. A search window is open, showing a search for 'Search' with various properties.

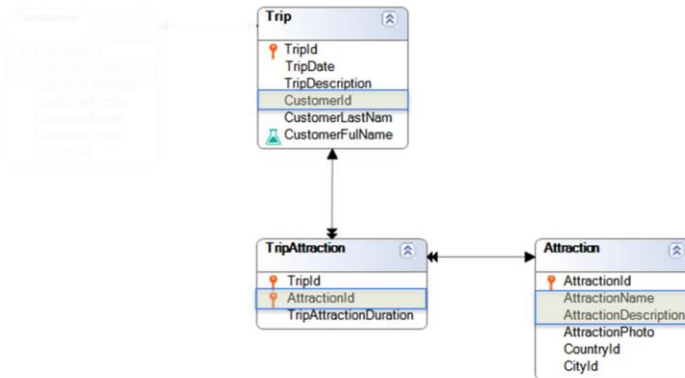
Se agora analisarmos a tabela base da parte variável, temos os atributos AttractionId, AttractionName e AttractionDescription como colunas do grid e não há atributos em nenhuma das propriedades do grid.

Estamos tentados a afirmar que a tabela base do grid seria ATTRACTION, mas não esqueçamos que os atributos da tab Conditions também devem ser considerados e ali encontramos o atributo CustomerId como parte de um filtro.

Portanto os atributos que encontramos são: AttractionId, AttractionName, AttractionDescription e CustomerId.

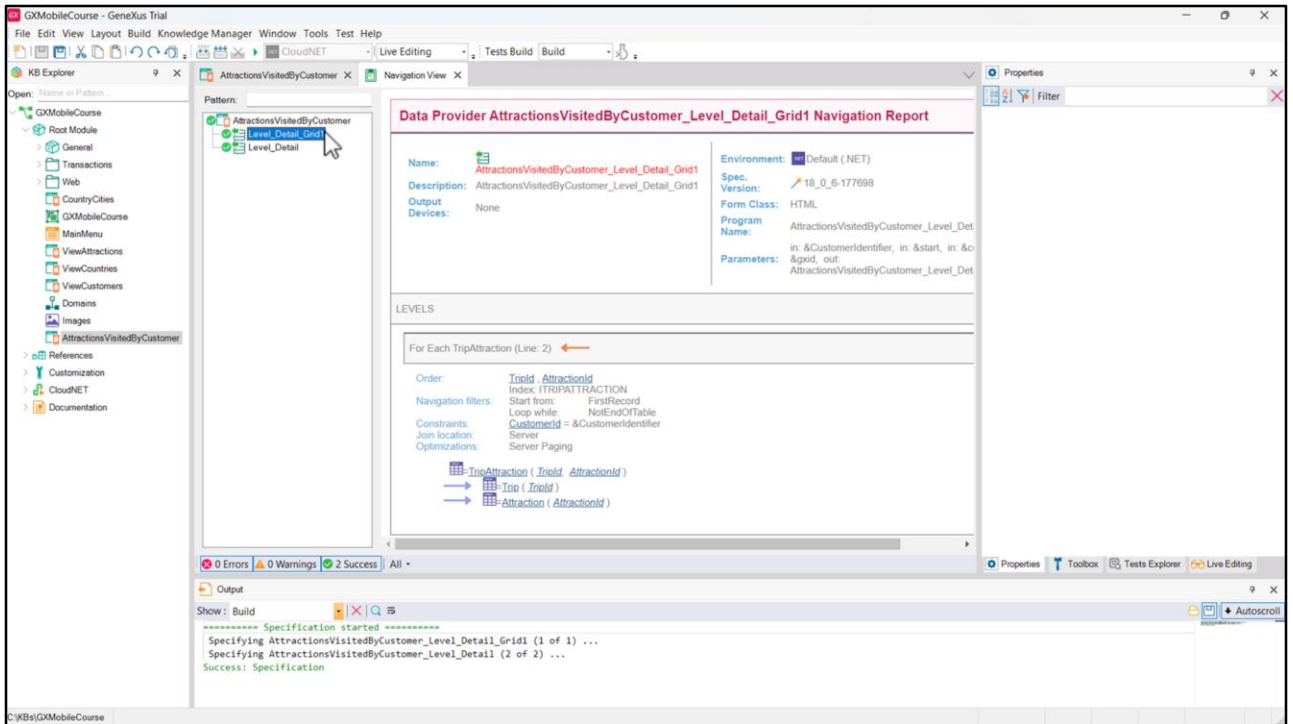


## Determining the base table of a variable part (grid)



Variable part base table: TRIPATTRACTION

Se analisarmos o diagrama de tabelas, vemos que a única tabela estendida que contém esses atributos é a tabela estendida de TripAttraction, portanto a tabela base da parte variável do Panel será TRIPATTRACTION.



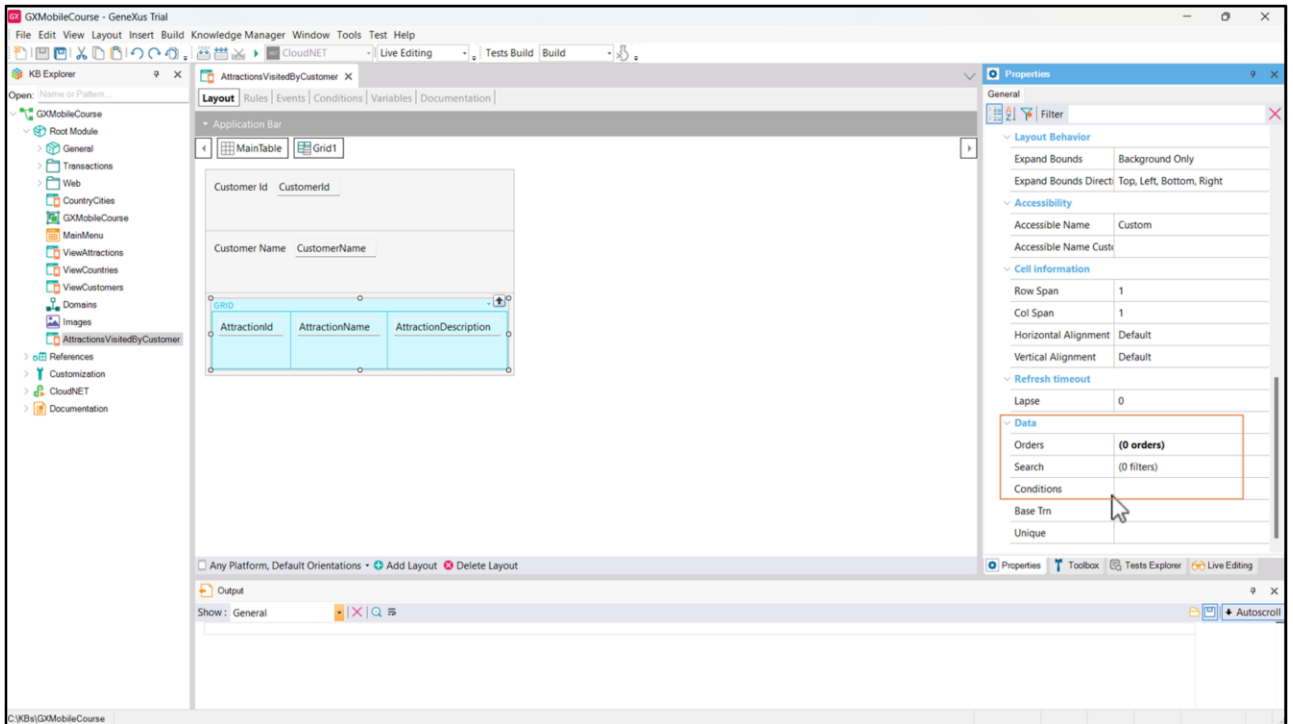
Se observarmos a lista de navegação do nó Level\_Detail\_Grid1 correspondente à parte variável, vemos que a tabela base é efetivamente TRIPATTRACTION, e que ela está filtrada pelo valor de CustomerId, devido à condição presente na tab Conditions.

Também vemos que são acessadas as tabelas Trip e Attraction para recuperar os dados dos atributos CustomerId, AttractionName e AttractionDescription.

Anteriormente mencionamos que o código presente nos eventos também é levado em consideração para a determinação das tabelas base da parte fixa e da parte variável. Veremos em outro vídeo os eventos de um objeto Panel.

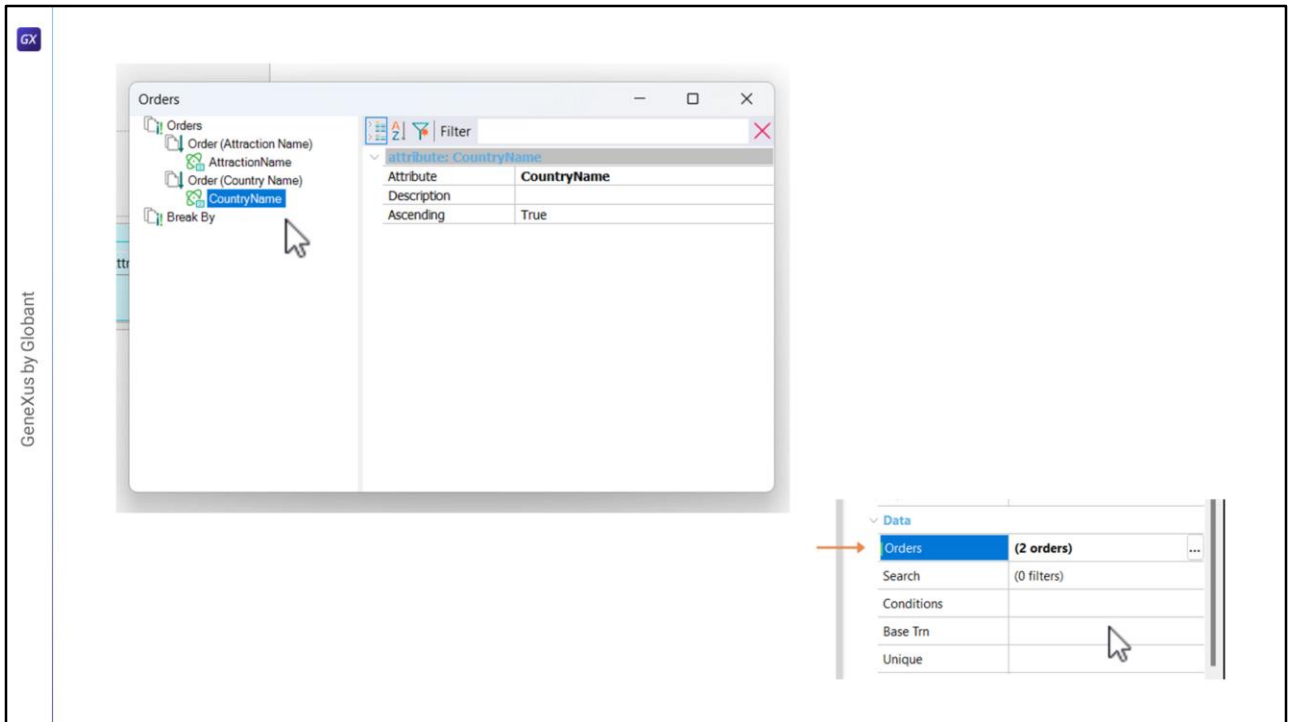
## Defining Orders, Searches & Conditions in Grids

Vejamos agora como podemos ordenar e agrupar a informação e como fazer pesquisas e estabelecer filtros e condições.



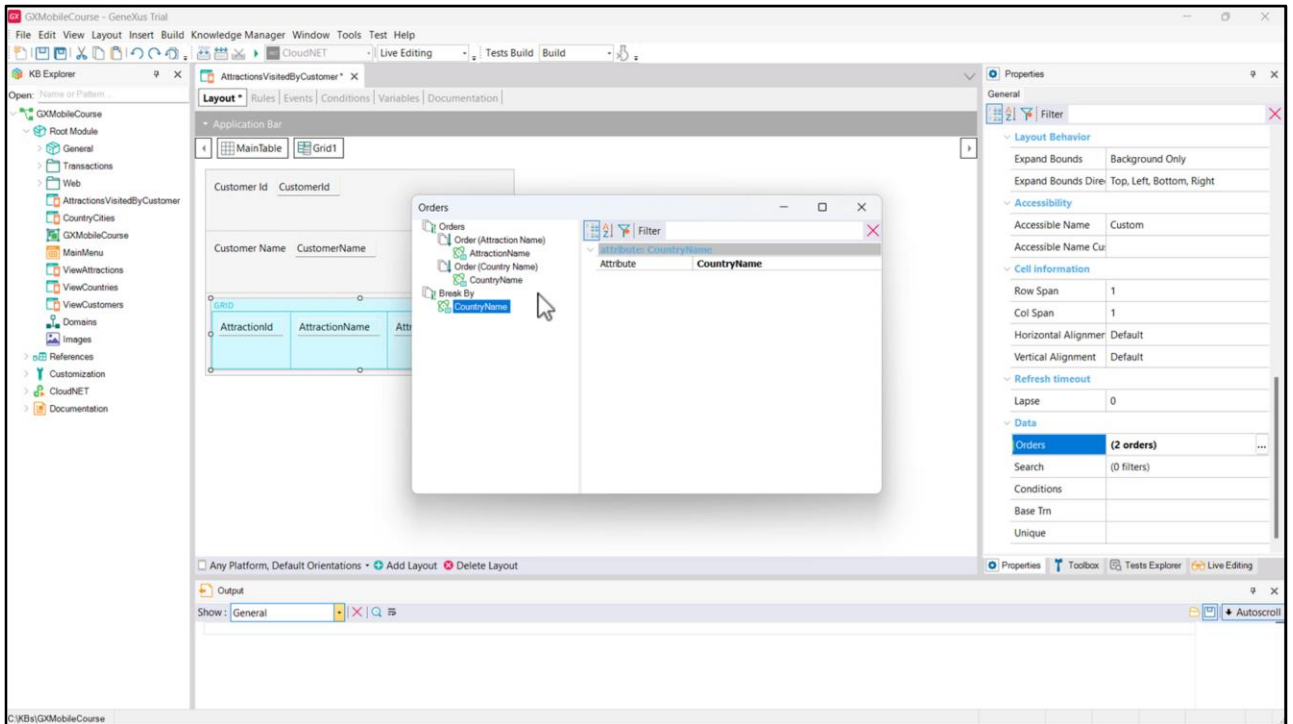
Uma característica fundamental quando apresentamos informação em grids é que ela aparece ordenada, segundo um ou vários critérios, de forma a ajudar o usuário a processar essa informação. Poderíamos, por exemplo, querer que a informação na lista de atrações apareça ordenada por nome, ou poderíamos querer ordená-la pelo nome do país em que se encontram.

Se formos às propriedades do grupo Data do grid, vemos que não foram definidas ordens, pesquisas nem condições para os dados.

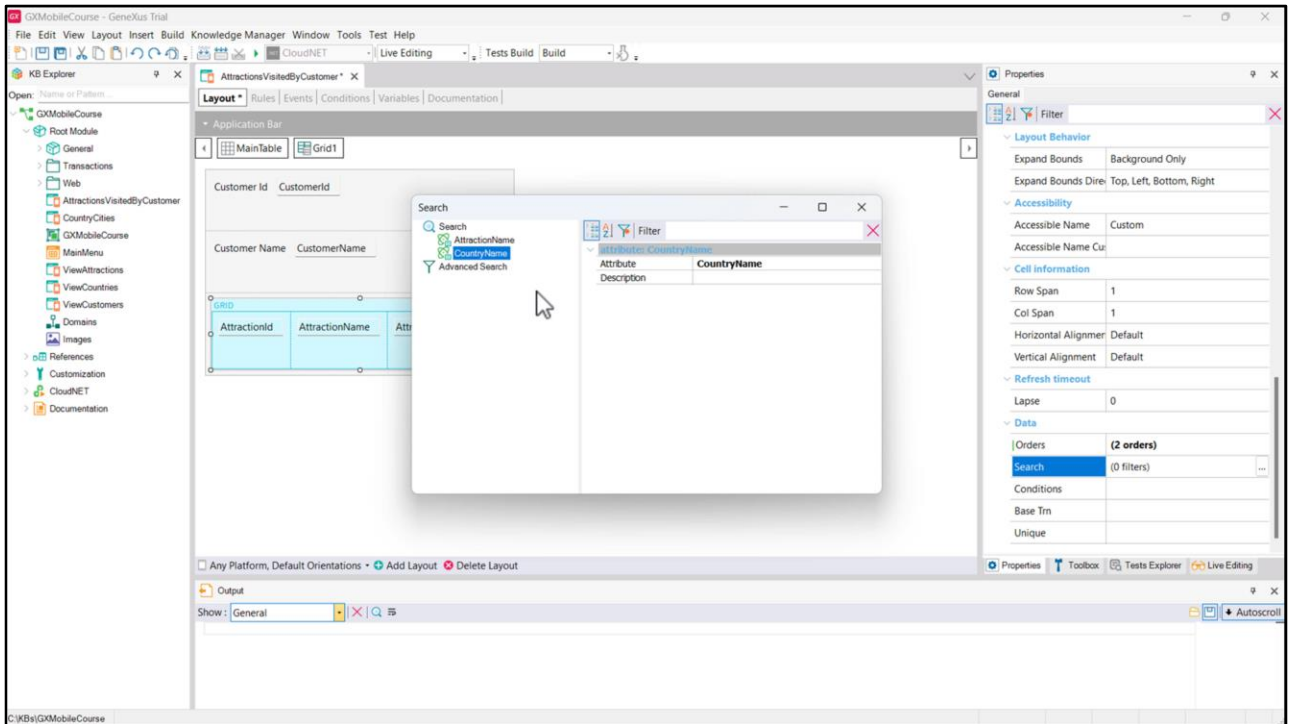


Para definir nossas ordens, abrimos o editor da propriedade Orders, clicamos com o botão direito, Add, Order, colocamos o nome, neste primeiro caso Attraction Name, e sobre ele clicamos com o botão direito, Add, Attribute: aqui escolhemos o atributo AttractionName. Criamos, da mesma forma, a ordem para classificar por nome de país: clique com o botão direito, Add, Order, colocamos o nome Country Name... e adicionamos o atributo CountryName.

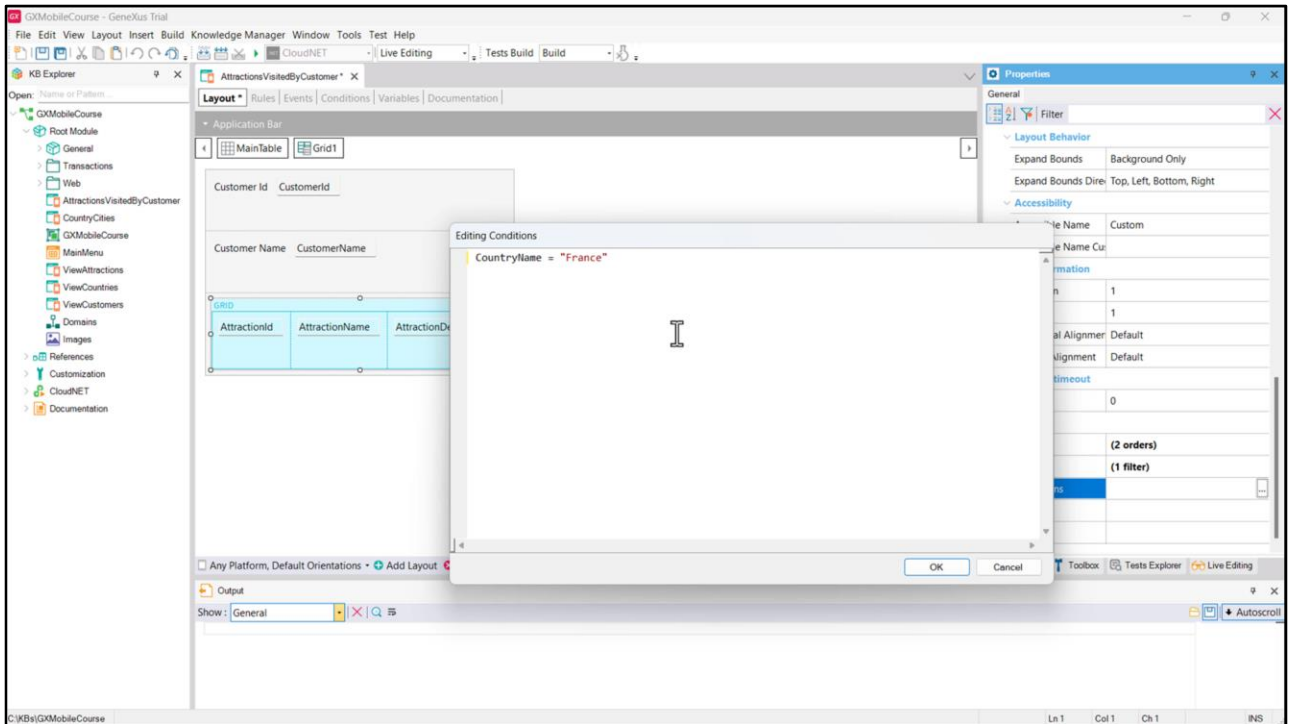
Vemos que agora aparecem 2 ordens definidas.



Em vez de querer ordenar por nome de país, é possível que o usuário também tenha interesse em ter a possibilidade de agrupar as atrações de acordo com o país ao qual pertencem. Voltamos à janela de definição de ordens, e no nó Break By clicamos com o botão direito, Add, Attribute e escolhemos o atributo CountryName para que faça o corte de controle.



Outra das funcionalidades fundamentais que toda aplicação deve oferecer são as pesquisas. Por exemplo, poderíamos pensar na possibilidade de pesquisar atrações por nome ou país. Para fazer isso, abrimos o editor da propriedade Search e simplesmente adicionamos os dois atributos pelos quais queremos pesquisar: AttractionName e CountryName.



Além disso, ocasionalmente pode ser necessário que alguns registros atendam a certas condições para serem exibidos: poderíamos querer que o grid de atrações mostre apenas atrações que estão na França, por exemplo, então na propriedade Conditions, escrevemos: `CountryName = "France"`.





Neste vídeo estudamos a lógica que emprega GeneXus na hora de carregar os dados na tela de um Panel e determinar suas tabelas base, e vimos como melhorar a experiência dos usuários de nossa aplicação oferecendo ordens, pesquisas e condições para os dados a serem mostrados.