

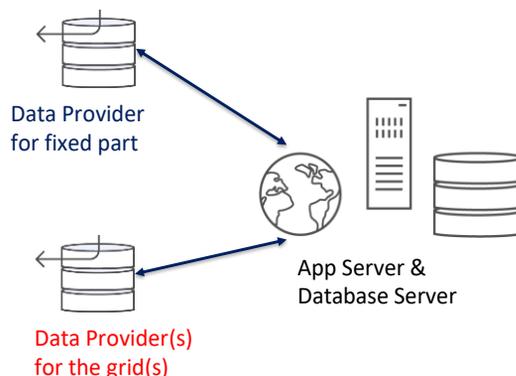
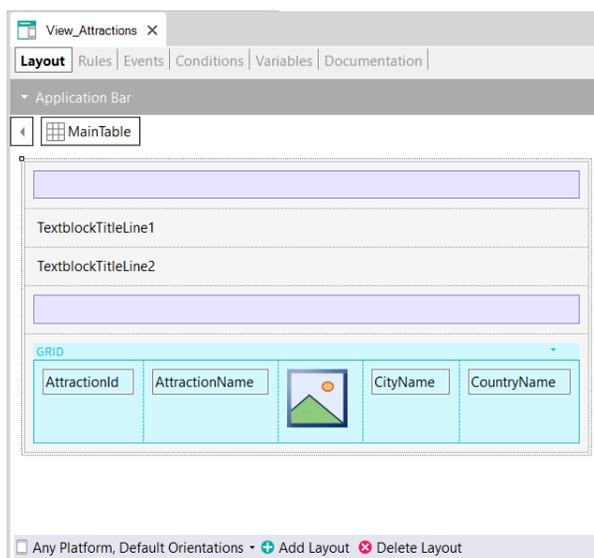
Determinação de tabelas base em um objeto panel



Um panel pode recuperar informações da base de dados automaticamente, apenas adicionando atributos em sua tela. Porém, saber como GeneXus determina quais tabelas deve acessar e a forma que as percorre é vital para que nossa aplicação funcione corretamente.

A seguir, estudaremos como são determinadas as tabelas base de um objeto panel.

Partes de um objeto panel



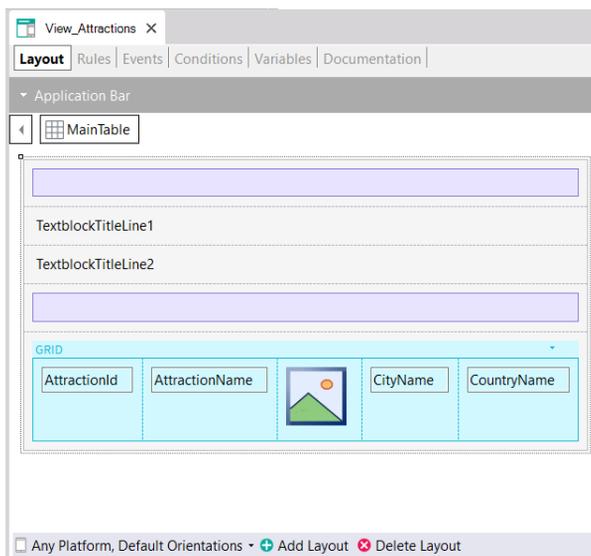
Como vimos anteriormente, um panel possui uma parte fixa onde estão os controles do form que não pertencem a um grid e uma ou mais partes variáveis, uma para cada grid que está no panel.

Para carregar os dados em cada parte do panel, GeneXus irá gerar automaticamente um data provider para cada parte que se encarregará de obter os dados da base de dados. Como estes data providers são independentes, cada um terá sua navegação própria, portanto cada parte do panel (fixa ou variável) terá sua própria tabela base independente das demais.

No exemplo, vemos elementos na parte fixa do panel e um grid, de modo que serão determinadas as tabelas base da parte fixa e da parte variável separadamente. Pode ocorrer 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 das partes a tenha.

É importante lembrar que neste, os panels funcionam de forma diferente dos webpanels, já que em um webpanel com um único grid, caso haja tabela base, a tabela base do webpanel é única e não existem duas tabelas base distintas separadas para a parte fixa e grid, como acontece com os panels.

Determinação de tabelas base de cada parte



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

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

Attributes involved in determining of **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

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

Para determinar a tabela base da parte fixa, serão levados em consideração os atributos pertencentes à parte fixa do form, os atributos pertencentes aos eventos associados à parte fixa, desde que estes 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 os da Application Bar.

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

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 como não visíveis, os atributos referenciados na Order, Search, Advanced Search e Conditions do grid, os atributos que pertencem ao código do evento Load sempre que estiverem fora das cláusulas For Each e presentes nos eventos de botões ou controles dentro do grid.

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

O grid também terá tabela base se sua propriedade Base Trn foi atribuída com uma transação base. Neste caso, os atributos que estão nas outras partes deverão pertencer à tabela estendida da tabela associada à transação base.

Determinação de tabelas base de cada parte

View_Attractions X

Layout Rules Events Conditions Variables Documentation

Application Bar

MainTable

TextblockTitleLine1

TextblockTitleLine2

GRID

AttractionId	AttractionName	CityName	CountryName

Any Platform, Default Orientations Add Layout Delete Layout

Fixed Part

Fixed Part: NO BASE TABLE

Variable part

Layout Rules Events **Conditions** Variables Documentation

1

Layout Rules **Events** Conditions Variables Documentation

Start

```

1 Event Start
2   TextblockTitleLine1.Caption = "The new age of"
3   TextblockTitleLine2.Caption = "EXPLORATION"
4 Endevent
5

```

Search

Advanced Search

Filter

Property	Value
search: Search	
Caption	
Option For Individual	False
Always Visible	False
Filter Operator	Contains
Case Sensitive	False
Break By	Platform Default

Data

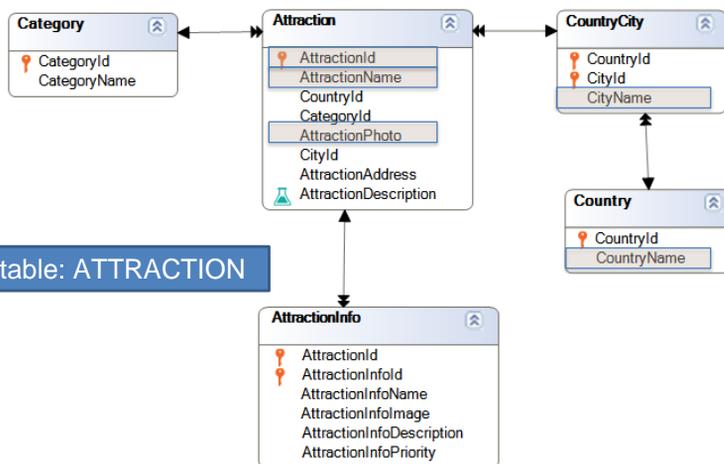
Component	Order
Orders	(0 order)
Search	(0 filters)
Conditions	
Base Trn	
Unique	

No exemplo que vemos, a parte fixa é composta apenas por dois textblocks no form e duas tabelas como separadores, 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, portanto a parte fixa deste panel não possui tabela base.

Se analisamos a parte variável, vemos que existem atributos nas colunas do grid, que são: AttractionId, AttractionName, AttractionPhoto, CityName e CountryName. Não há atributos nas propriedades Order, Search, Conditions do grid e também não está atribuído o valor da propriedade Base Trn. E não há atributos no evento Load ou em eventos de controles dentro do grid.

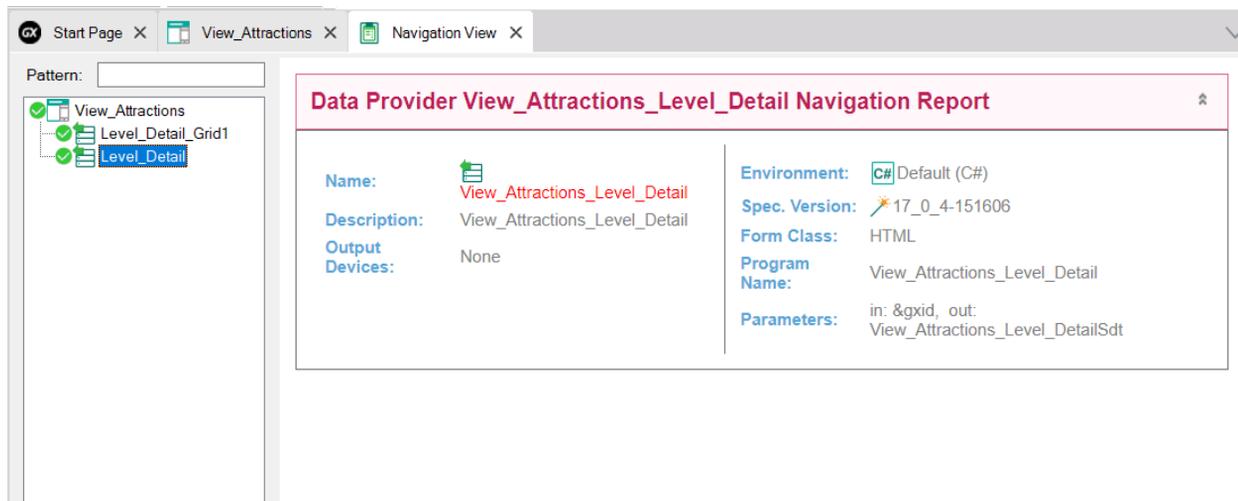
Determinação de tabelas base de cada parte

- Attributes in the grid: AttractionId, AttractionName, AttractionPhoto, CityName y CountryName



Para determinar a tabela base, devemos buscar todas as tabelas estendidas que contenham os atributos mencionados. Vemos que a única que contém todos eles é a tabela estendida de Attraction, pois atinge as tabelas Attraction, CountryCity, Country e Category e nas 3 primeiras estão todos os atributos que procurávamos. 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, que percorrerá a tabela Attraction e recuperará os dados que carregarão o grid.

Determinação de tabelas base de cada parte



Se clicamos com o botão direito do mouse sobre o nome do panel e selecionamos View Navigation, vemos a lista de navegação do panel.

O nó Level_Detail correspondente à parte fixa está vazio, portanto, confirmamos que a parte fixa não possui tabela base.

Determinação de tabelas base de cada parte

Pattern:

View_Attractions
 Level_Detail_Grid1
 Level_Detail

Data Provider View_Attractions_Level_Detail_Grid1 Navigation Report

Name: View_Attractions_Level_Detail_Grid1
Description: View_Attractions_Level_Detail_Grid1
Output: None
Devices: None

Environment: Default (C#)
Spec: 17_0_4-151606
Version: 17_0_4-151606
Form Class: HTML
Program Name: View_Attractions_Level_Detail_Grid1
Parameters: in: &start, in: &count, in: &gxid, out: View_Attractions_Level_Detail_Grid1Sdt

LEVELS

For Each Attraction (Line: 2)

Order: [AttractionId](#)
Index: IATTRACTION
Navigation filters: Start from: FirstRecord
 Loop while: NotEndOfTable
Join location: Server
Optimizations: Server Paging

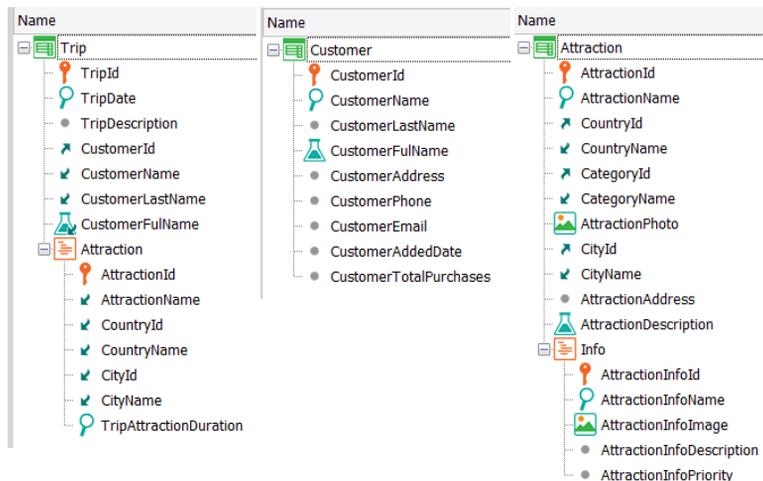
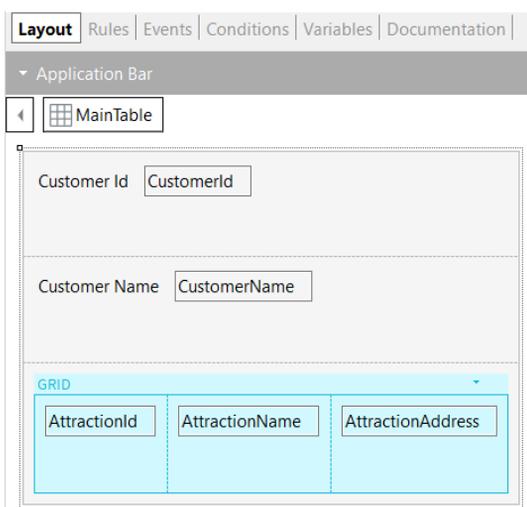
[Attraction \(AttractionId \)](#) INTO [AttractionPhoto Uri](#)
[AttractionId](#) [AttractionName](#) [AttractionPhoto](#) [CountryId](#) [CityId](#)
[Country \(CountryId \)](#) INTO [CountryName](#)
[CountryCity \(CountryId, CityId \)](#) INTO [CityName](#)

Se agora selecionamos o nó correspondente ao Grid1, vemos que na lista de navegação do data provider correspondente diz For Each Attraction, então a tabela base é efetivamente ATTRACTION, como havíamos deduzido anteriormente.

Mais abaixo vemos que acessa a tabela Attraction para recuperar os dados de AttractionId, AttractionName, AttractionPhoto, CountryId e CityId e então com o CountryId obtido acessa a tabela Country para obter o CountryName e com o CountryId e CityId acessa a tabela CountryCity para trazer o CityName.

Em resumo, vemos que só pelo fato de que o grid possua atributos, neste caso em suas colunas, GeneXus foi capaz de determinar automaticamente quais tabelas devia percorrer e criar tudo o que fosse necessário para recuperar os dados requeridos a partir da base de dados.

Outro exemplo de determinação de tabelas base



Vejamos outro exemplo de determinação de tabelas base. A agência de viagens quer ter um panel onde, para um determinado cliente recebido por parâmetro, sejam mostradas as atrações visitadas por esse cliente.

Essa informação está modelada na transação Trip onde cada viagem tem um cliente (observemos que CustomerId é chave estrangeira e os atributos CustomerName, CustomerLastName e CustomerFullName são inferidos) e cada viagem tem também muitas atrações que são visitadas, representadas pelo segundo nível da transação Trip. Também são exibidas a transação Customer que tem os dados dos clientes e a transação Attraction com os dados das atrações.

Determinação de tabela base da parte fixa

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

Application Bar

MainTable

Customer Id CustomerId

Customer Name CustomerName

GRID

AttractionId AttractionName AttractionAddress

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

```
1 Parm(in: &CustoEmerIdentifier);
```

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

```
1 CustomerId = &CustomerIdentifier;
2
```

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

Events

```
1
```

Customer

- CustomerId
- CustomerName
- CustomerLastName
- CustomerFullName
- CustomerAddress
- CustomerPhone
- CustomerEmail
- CustomerAddedDate
- CustomerTotalPurchases

Fixed part base table: CUSTOMER

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

Isso 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.

Determinação de tabela base da parte fixa

Pattern:

AttractionsVisitedByCustomer
 Level_Detail_Grid1
 Level_Detail

Data Provider AttractionsVisitedByCustomer_Level_Detail Navigation Report

Name: AttractionsVisitedByCustomer_Level_Detail
Description: AttractionsVisitedByCustomer_Level_Detail
Output: None
Devices: None

Environment: Default (C#)
Spec. Version: 17_0_4-151606
Form Class: HTML
Program Name: AttractionsVisitedByCustomer_Level_Detail
Parameters: in: &CustomerId, in: &gxid, out: AttractionsVisitedByCustomer_Level_DetailSdt

LEVELS

For First Customer (Line: 1)

Order: CustomerId
 Index: ICUSTOMER
Navigation filters: Start from: CustomerId = &CustomerId
 Loop while: CustomerId = &CustomerId
Optimizations: First 1 record(s)

[-] Customer (CustomerId) INTO CustomerId CustomerName

0 Errors 0 Warnings 2 Success All

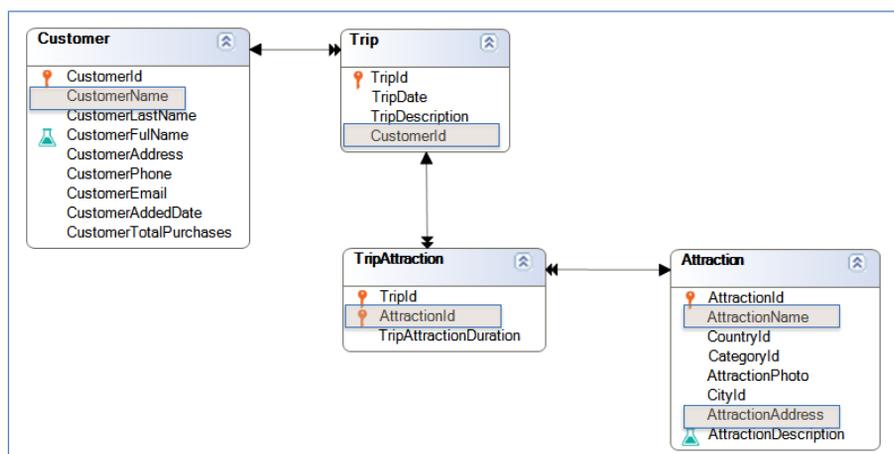
Se observamos a lista de navegação do nó Level_Detail correspondente à parte fixa, vemos que a tabela base é Customer, como havíamos determinado antes e que é acessada a referida tabela para recuperar os dados dos atributos CustomerId e CustomerName.

Determinação de tabela base da parte variável (grid)

The screenshot illustrates the configuration of a variable part (grid) in GeneXus. The main table 'MainTable' contains fields 'Customer Id' and 'Customer Name'. A grid is defined with columns 'AttractionId', 'AttractionName', and 'AttractionAddress'. The 'Rules' tab shows a parameter definition: `1 Parm(in: &CustoEmerIdentifier);`. The 'Conditions' tab shows a condition: `1 CustomerId = &CustomerIdentifier;`. The 'Events' tab shows an event definition: `1`. A 'Search' dialog is open, showing the 'Data' tab with a list of tables: 'Orders (0 orders)', 'Search (0 filters)', 'Conditions', 'Base Trn', and 'Unique'. The 'Search' table is selected.

Se agora analisamos a tabela base da parte variável, temos os atributos `AttractionId`, `AttractionName` e `AttractionAddress` 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 vamos esquecer que os atributos da tab `Conditions` também devem ser considerados e ali encontramos o atributo `CustomerId` como parte de um filtro.

Determinação de tabela base da parte variável (grid)



Variable part base table: TRIPATTRACTION

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

Se analisamos o diagrama de tabelas, vemos que a única tabela estendida que contém estes atributos é a tabela estendida de TripAttraction, então a tabela base da parte variável do panel será TRIPATTRACTION.

Determinação de tabela base da parte variável (grid)

Data Provider AttractionsVisitedByCustomer_Level_Detail_Grid1 Navigation Report

Name: AttractionsVisitedByCustomer_Level_Detail_Grid1
Description: AttractionsVisitedByCustomer_Level_Detail_Grid1
Output Devices: None

Environment: Default (C#)
Spec. Version: 17_0_4-151606
Form Class: HTML
Program Name: AttractionsVisitedByCustomer_Level_Detail_G
Parameters: in: &CustomerId, in: &start, in: &count, &gid, out: AttractionsVisitedByCustomer_Level_Detail_G

LEVELS

For Each TripAttraction (Line: 2)

Order: Tripld , AttractionId
Index: ITRIPATTRACTION

Navigation filters: Start from: FirstRecord
 Loop while: NotEndOfTable

Constraints: CustomerId = &CustomerId
Join location: Server
Optimizations: Server Paging

	=TripAttraction (<u>Tripld , AttractionId</u>) INTO AttractionId Tripld
	=Trip (<u>Tripld</u>) INTO CustomerId
	=Attraction (<u>AttractionId</u>) INTO AttractionName AttractionAddress

0 Errors 0 Warnings 2 Success All -

Se observamos a lista de navegação do nó Level_Detail_Grid1 correspondente à parte variável, vemos que a tabela base é efetivamente TRIPATTRACTION, e que a mesma é 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 AttractionAddress respectivamente.

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. Vamos ver em outro vídeo os eventos de um objeto panel.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications