

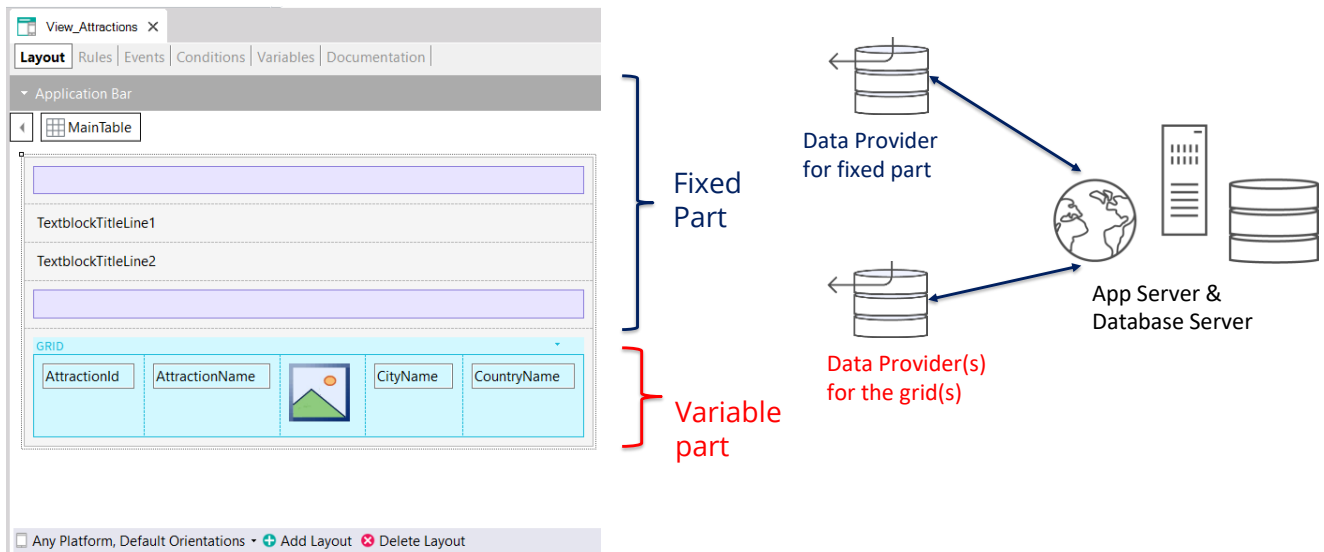
Determining the base tables in a panel object



A panel can automatically retrieve information from the database by simply adding attributes to its display. However, knowing how GeneXus determines which tables to access and how it runs through them is crucial for our application to work correctly.

Next, we will study how to determine the base tables of a panel object.

Parts of a panel object



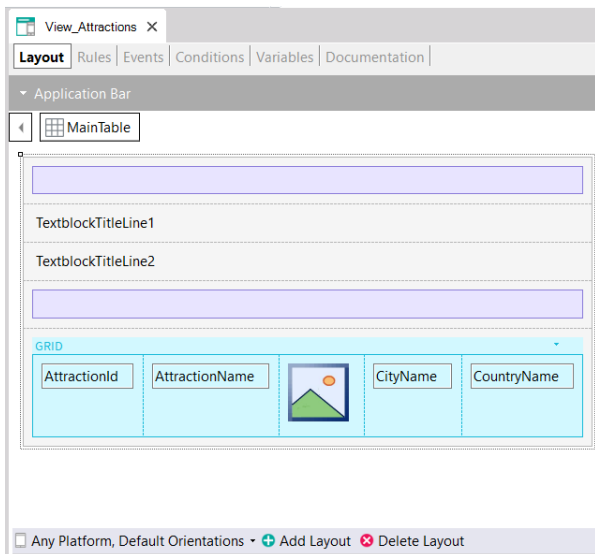
As we saw before, a panel has a fixed part where the form controls that don't belong to a grid are located; also, it has one or more variable parts, one for each grid found in the panel.

To load the data in each part of the panel, GeneXus will automatically generate a data provider for each part that will be in charge of obtaining the data from the database. As these data providers are independent, each one will have its own navigation, so each part of the panel (fixed or variable) will have its own base table that will be independent of the others.

In the example we see elements in the fixed part of the panel and a grid, so the base tables of the fixed part and the variable part will be determined separately. It may happen that the fixed part has a base table and the grid doesn't; that the grid has a base table and the fixed part doesn't; that both parts have a base table, or that neither part has a base table.

Remember that here panels work differently than web panels. In a web panel with a single grid, if there is a base table, the base table of the web panel is unique and there aren't two separate base tables for the fixed part and the grid, as in the case of panels.

Determining the base tables of each 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

Since in a panel object the fixed part and the grid determine independent navigations and each part will have its own base table, it's as if there were two parallel For Each commands.

To determine the base table of the fixed part, the attributes that belong to the fixed part of the form, the attributes that belong to the events associated with the fixed part, will be taken into account as long as these attributes are outside a For Each command.

These events are the Refresh event and the events associated with buttons or controls of the fixed part, including those of the Application Bar.

In addition, the attributes of the Conditions Tab of the Panel object must also be taken into account to determine the base table of the fixed part.

To determine the base table of the variable part, in this case of the grid, the attributes taken into account will be those included in the grid columns, both visible and hidden; the attributes referenced in the Order, Search, Advanced Search and Conditions of the grid; and the attributes belonging to the Load event code as long as they are outside For Each clauses and included in the events of buttons or controls within the grid.

The attributes in the Conditions tab will also be taken into account for determining the base table of all the grids included in the panel.

Also, the grid will have a base table if its Base Trn property was assigned with a base transaction. In this case, the attributes that are in the other parts must belong to the extended table of the table associated with the base transaction.

Determining the base tables of each part

The screenshot shows the 'View_Attractions' panel in the GeneXus IDE. The panel is divided into two main sections:

- Fixed Part:** Indicated by a blue bracket on the right. It contains two text blocks (TextblockTitleLine1 and TextblockTitleLine2) and two tables (MainTable and another unnamed table).
- Variable part:** Indicated by a red bracket on the right. It contains a grid with columns for AttractionId, AttractionName, CityName, and CountryName.

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

```

Fixed Part: NO BASE TABLE

The screenshot shows the 'Search' panel in the GeneXus IDE. The panel is divided into two main sections:

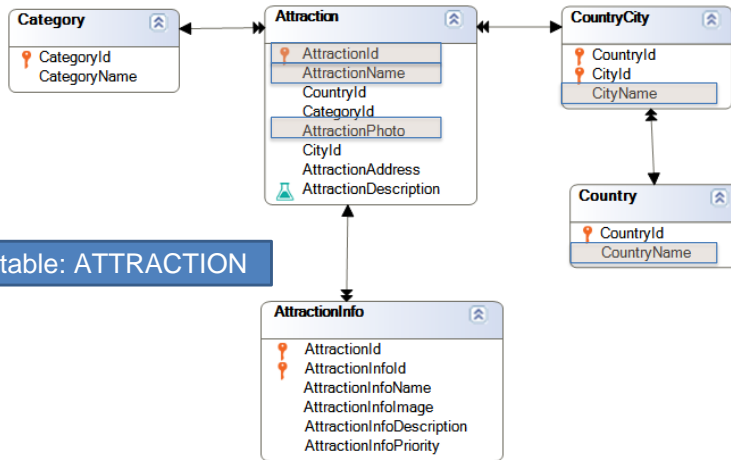
- Fixed Part:** Indicated by a blue bracket on the right. It contains two text blocks (TextblockTitleLine1 and TextblockTitleLine2) and two tables (MainTable and another unnamed table).
- Variable part:** Indicated by a red bracket on the right. It contains a grid with columns for AttractionId, AttractionName, CityName, and CountryName.

In the example shown, the fixed part is composed only of two textblocks in the form and two tables as separators; i.e. there are no attributes. In addition, there are no attributes in the Conditions tab of the panel. There is no Refresh event and there are no form control or button events in the ApplicationBar; so, the fixed part of this panel has no base table.

If we analyze the variable part, we see that the grid columns contain the attributes AttractionId, AttractionName, AttractionPhoto, CityName, and CountryName. There are no attributes in the Order, Search, Conditions properties of the grid and the value of the Base Trn property is not assigned either. Also, there are no attributes in the Load event nor in events of controls inside the grid.

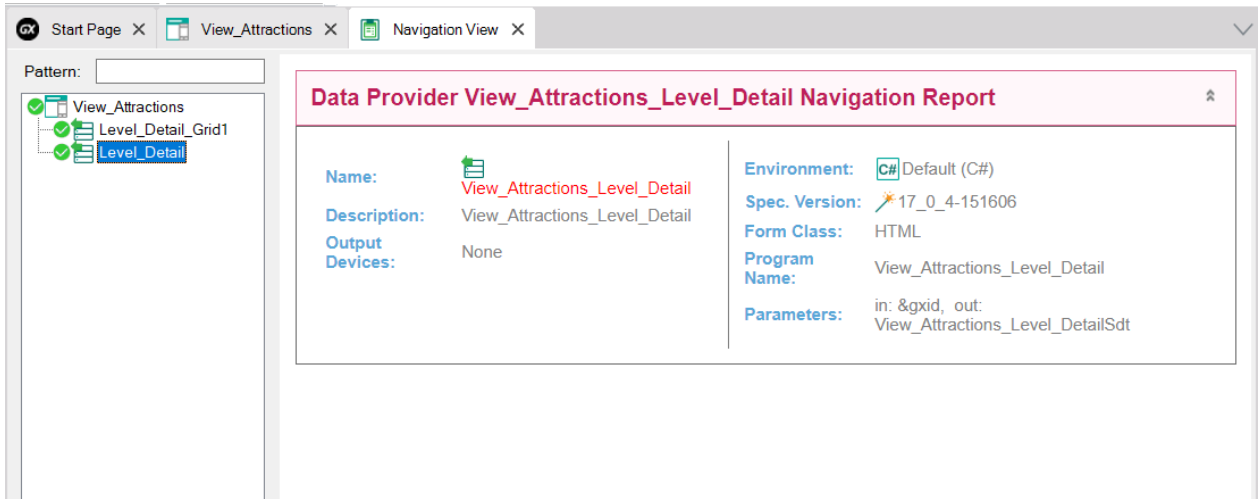
Determining the base tables of each part

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



To determine the base table, we must find all the extended tables that contain the attributes mentioned above. Note that the only one that contains all of them is the extended table of Attraction, since it reaches the tables Attraction, CountryCity, Country and Category and the first 3 contain all the attributes we were looking for. Therefore, the variable part of the panel, composed by the grid, will have the Attraction base table. This means that GeneXus will build a data provider as a service on the back end, which will run through the Attraction table and retrieve the data that the grid will load.

Determining the base tables of each part



If we right-click on the panel name and select View Navigation, we see the panel navigation list.

The Level_Detail node corresponding to the fixed part is empty, so we confirm that the fixed part has no base table.

Determining the base tables of each part

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

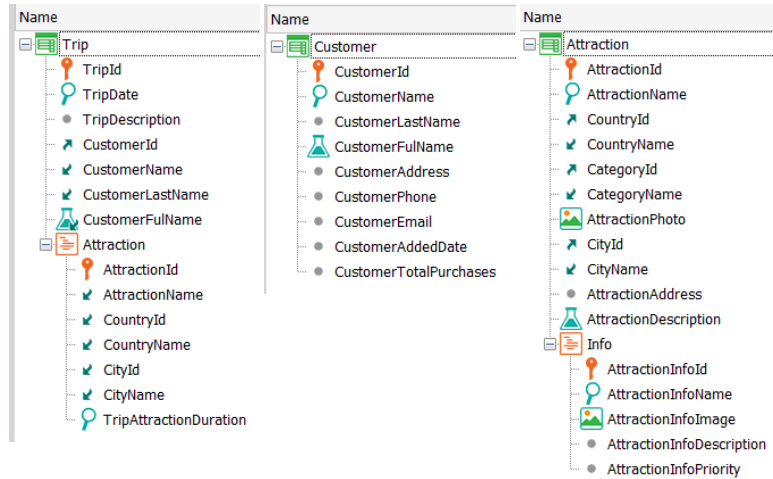
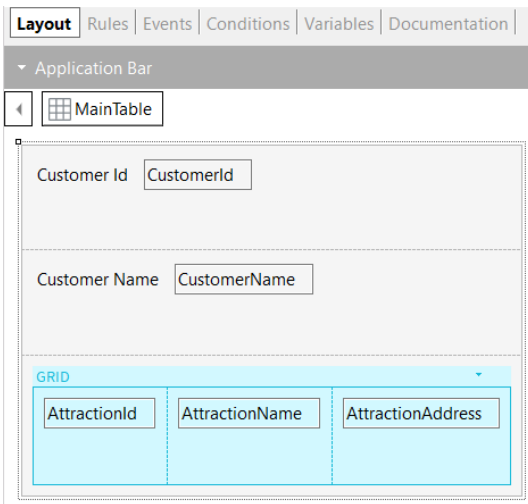
	-Attraction (AttractionId) INTO AttractionPhoto Uri
	AttractionId AttractionName AttractionPhoto CountryId CityId
	-Country (CountryId) INTO CountryName
	-CountryCity (CountryId CityId) INTO CityName

If we now select the node corresponding to Grid1, we see that the navigation list of the corresponding data provider says For Each Attraction, so the base table is indeed ATTRACTION, as we had previously inferred.

Below we see that it accesses the Attraction table to retrieve the AttractionId, AttractionName, AttractionPhoto, CountryId and CityId data; then, with the obtained CountryId it accesses the Country table to obtain the CountryName, and with CountryId and CityId it accesses the CountryCity table to get the CityName.

In summary, we see that just because the grid has attributes, in this case in its columns, GeneXus was able to automatically determine which tables to run through and create everything necessary to retrieve the required data from the database.

Another example of base table determination



Let's see another example of base table determination. The travel agency wants to have a panel that shows the attractions visited by a certain customer received by parameter.

This information is modeled in the Trip transaction where each trip has a customer (note that CustomerId is a foreign key and the attributes CustomerName, CustomerLastName and CustomerFullName are inferred), and each trip also has many attractions that are visited, represented by the second level of the Trip transaction. The Customer transaction containing the customers' data and the Attraction transaction containing data on attractions are also shown.

Determining the base table of a fixed part

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

- Customer Id (CustomerId)
- Customer Name (CustomerName)
- GRID (containing AttractionId, AttractionName, and AttractionAddress)

The Rules tab shows the following code:

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

The Conditions tab shows the following code:

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

The Events tab is empty.

The 'Customer' table is shown with the following fields:

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

Fixed part base table: CUSTOMER

First, let's analyze the fixed part of the panel. In addition to the attributes included in the form, if we look for attributes in other parts of the panel, we see that the rules include only the &CustomerIdentifier variable that receives the identifier of the customer to be displayed; in the Conditions tab, we find a filter that ensures that only the data of the customer received by parameter will be displayed. We have nothing in the events tab.

This means that the only attributes to be analyzed for the fixed part are CustomerId (found in the form and in the Conditions tab) and CustomerName that is found in the form.

The base table is then CUSTOMER as it contains both attributes.

Determining the base table of a fixed part

The screenshot displays the GeneXus IDE interface. On the left, a tree view shows the project structure with 'AttractionsVisitedByCustomer' expanded to show 'Level_Detail_Grid1' and 'Level_Detail'. The main window shows the 'Data Provider AttractionsVisitedByCustomer_Level_Detail Navigation Report'. The report includes the following information:

- Name:** AttractionsVisitedByCustomer_Level_Detail
- Description:** AttractionsVisitedByCustomer_Level_Detail
- Output Devices:** None
- Environment:** C# Default (C#)
- Spec. Version:** 17_0_4-151606
- Form Class:** HTML
- Program Name:** AttractionsVisitedByCustomer_Level_Detail
- Parameters:** in: &CustomerIdIdentifier, in: &gxid, out: AttractionsVisitedByCustomer_Level_DetailSdt

The report also shows the 'LEVELS' section, which includes a 'For First Customer (Line: 1)' section with the following details:

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

The SQL query for this level is: `=Customer (CustomerId) INTO CustomerId CustomerName`. At the bottom of the IDE, a status bar indicates 0 Errors, 0 Warnings, and 2 Success.

If we look at the navigation list of the Level_Detail node corresponding to the fixed part, we see that the base table is Customer, as we had determined before, and that this table is accessed to retrieve the data of the CustomerId and CustomerName attributes.

Determining the base table of a variable part (grid)

The screenshot displays the GeneXus IDE interface. On the left, a 'MainTable' component is shown with fields for 'Customer Id' (CustomerId), 'Customer Name' (CustomerName), and a 'GRID' component. The 'GRID' component has three columns: 'AttractionId', 'AttractionName', and 'AttractionAddress'. A blue arrow points from the 'GRID' component to the 'Rules' tab of the code editor.

The 'Rules' tab shows the following code:

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

The 'Conditions' tab shows the following code:

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

The 'Events' tab shows the following code:

```
1
```

Below the code editor, a 'Search' dialog box is open, showing a list of search results. The 'Data' tab is selected, showing the following results:

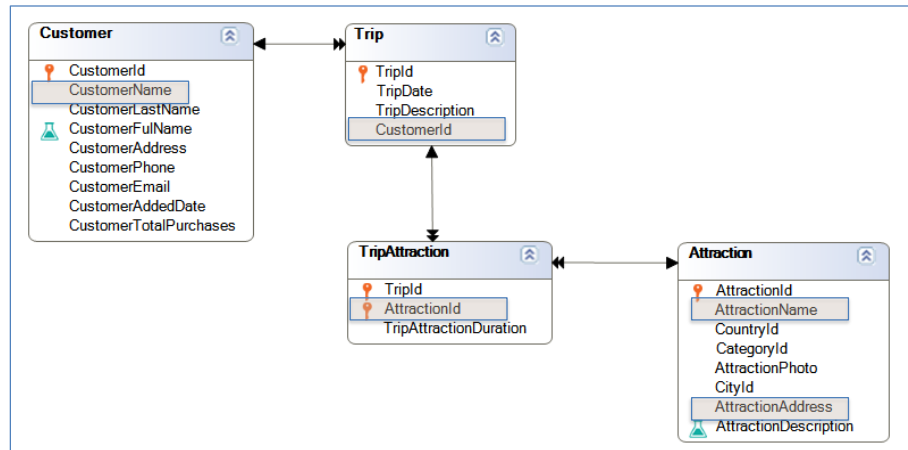
Data	Count
Orders	(0 orders)
Search	(0 filters)
Conditions	
Base Trn	
Unique	

The 'Search' dialog box also shows a 'Filter' tab with the following properties:

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

If now we analyze the base table of the variable part, we find the attributes AttractionId, AttractionName and AttractionAddress as grid columns, and that there are no attributes in any of the grid properties. We may be tempted to say that the base table of the grid would be ATTRACTION, but let's not forget that the attributes of the Conditions tab must also be considered and there we find the CustomerId attribute as part of a filter.

Determining the base table of a variable part (grid)



Variable part base table: TRIPATTRACTION

Therefore, the attributes we find are: AttractionId, AttractionName, AttractionAddress, CustomerId, and CustomerName.

If we analyze the table diagram, we see that the only extended table containing these attributes is the extended table of TripAttraction, so the base table of the variable part of the panel will be TRIPATTRACTION.

Determining the base table of a variable part (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, &xid, out: AttractionsVisitedByCustomer_Level_Detail_G

LEVELS

For Each TripAttraction (Line: 2)

Order: TripId , AttractionId
Index: ITRIPATTRACTION
Navigation filters: Start from: FirstRecord
 Loop while: NotEndOfTable
Constraints: CustomerId = &CustomerId
Join location: Server
Optimizations: Server Paging

TripAttraction (TripId , AttractionId) INTO AttractionId TripId
 Trip (TripId) INTO CustomerId
 Attraction (AttractionId) INTO AttractionName AttractionAddress

0 Errors 0 Warnings 2 Success All

In the navigation list of the Level_Detail_Grid1 node corresponding to the variable part, note that the base table is indeed TRIPATTRACTION, and that it is filtered by the CustomerId value, due to the condition included in the Conditions tab.

We also see that the Trip and Attraction tables are accessed to retrieve data from the CustomerId, AttractionName and AttractionAddress attributes, respectively.

We have mentioned before that the code found in the events is also taken into account for determining the base tables of the fixed part and variable part. Let's see in another video the events of a panel object.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications