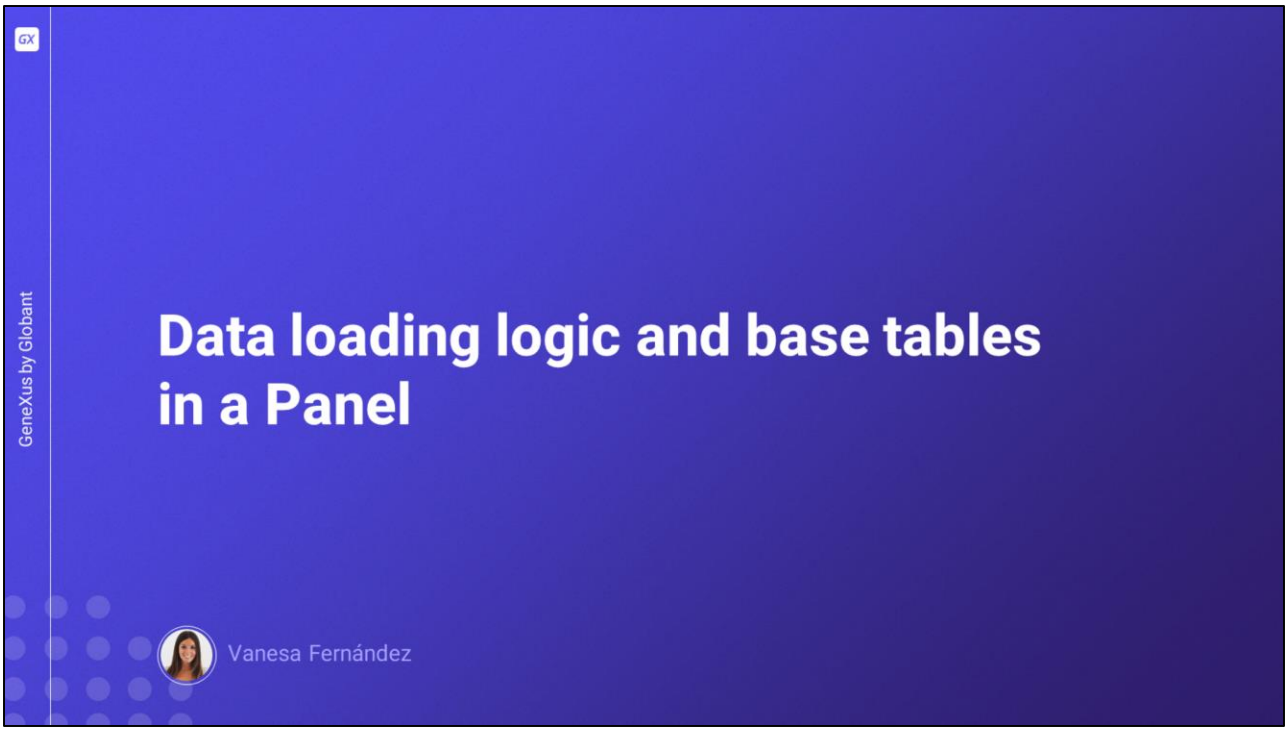
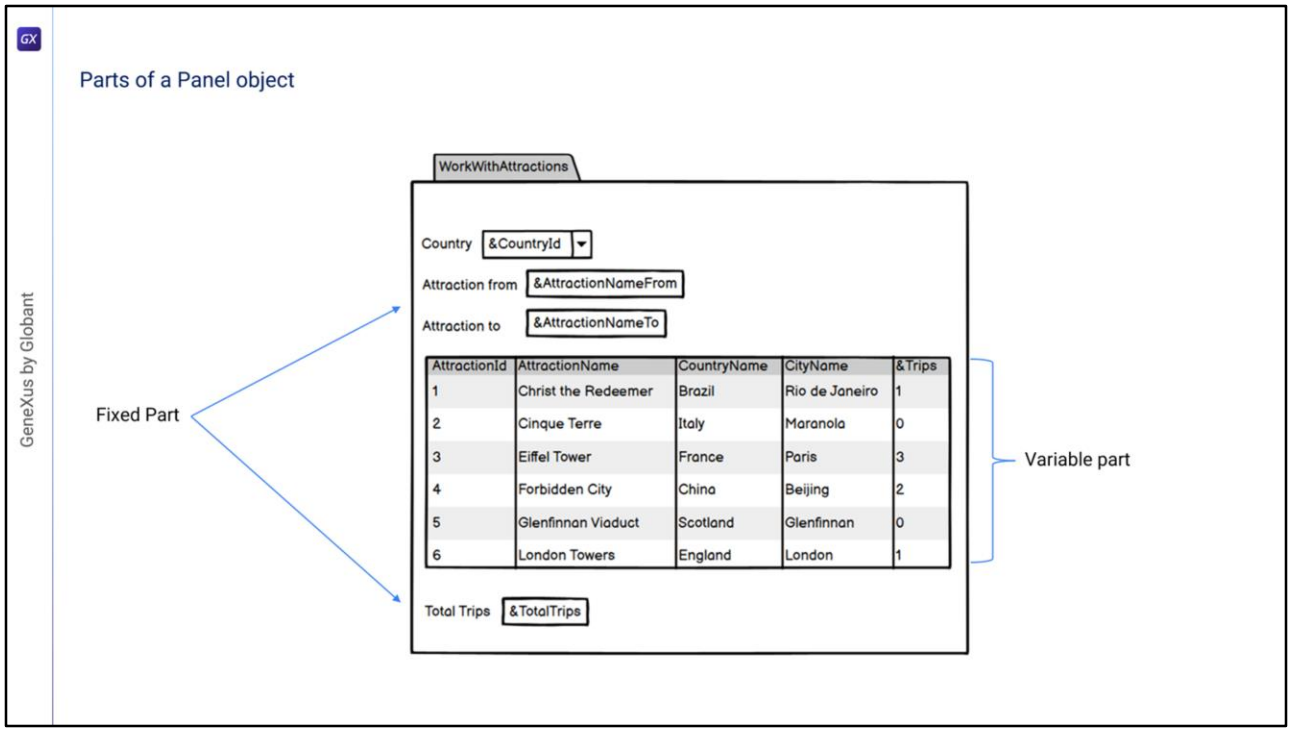


En este video estudiaremos cómo se carga la pantalla de un objeto Panel dependiendo de los componentes de interfaz gráfica presentes en ella, cómo son determinadas sus tablas base y cómo hacemos para buscar, ordenar y filtrar la información en pantalla.



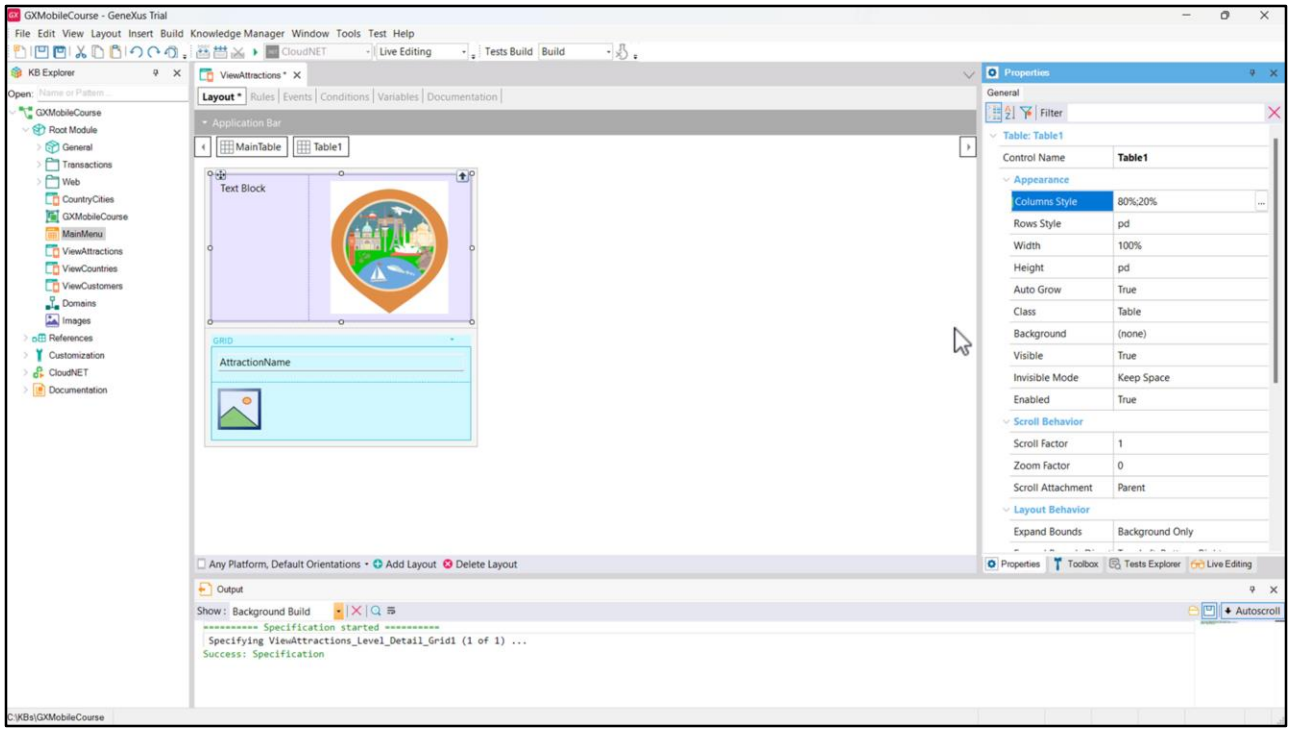
En un objeto Panel podemos identificar dos partes distintas:

llamamos parte fija o plana a la parte que contiene todo aquello que esté en el form y que no esté incluido en ningún grid. Esto comprende a controles como textblocks, combos, botones y otros que podemos arrastrar desde la barra de herramientas.

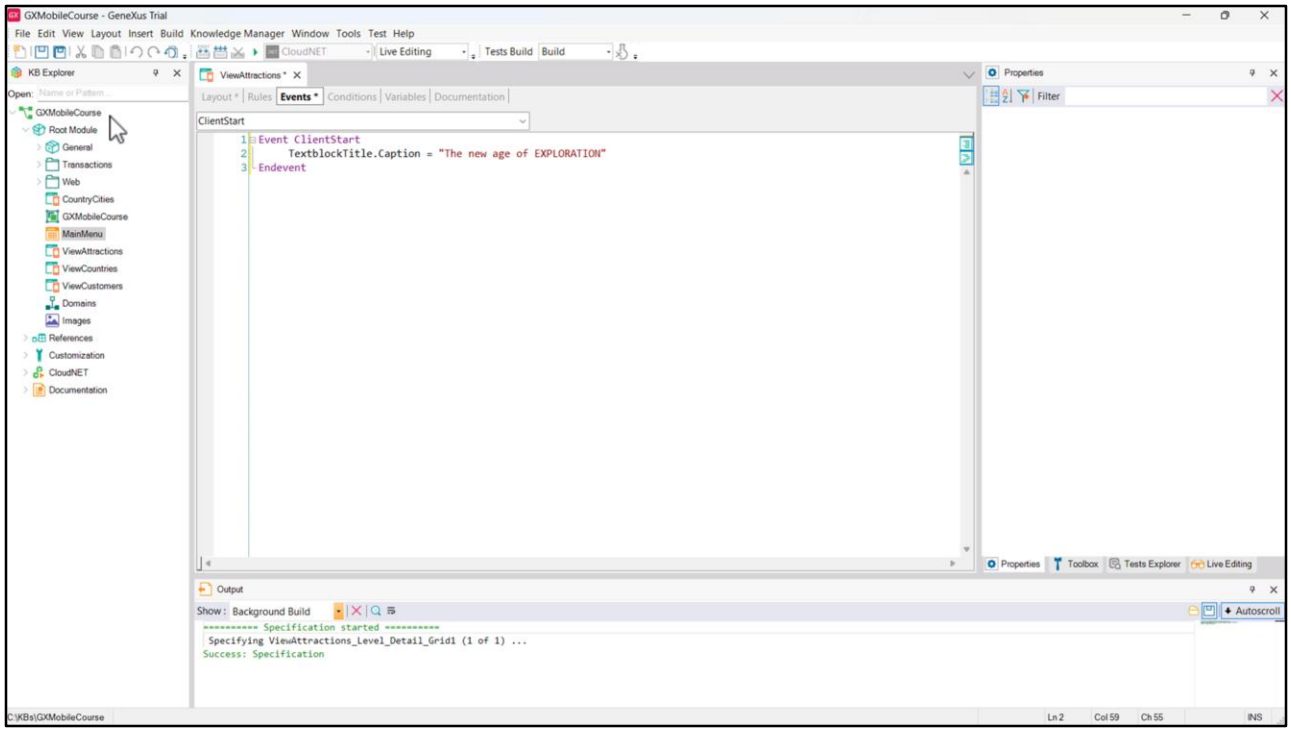
La segunda parte la denominaremos variable, y está compuesta por el o los grids que hayan sido incluidos en el form.

En un objeto Panel siempre tendremos una parte fija, y podemos tener una parte variable por cada una de las grillas que tenga el Panel.

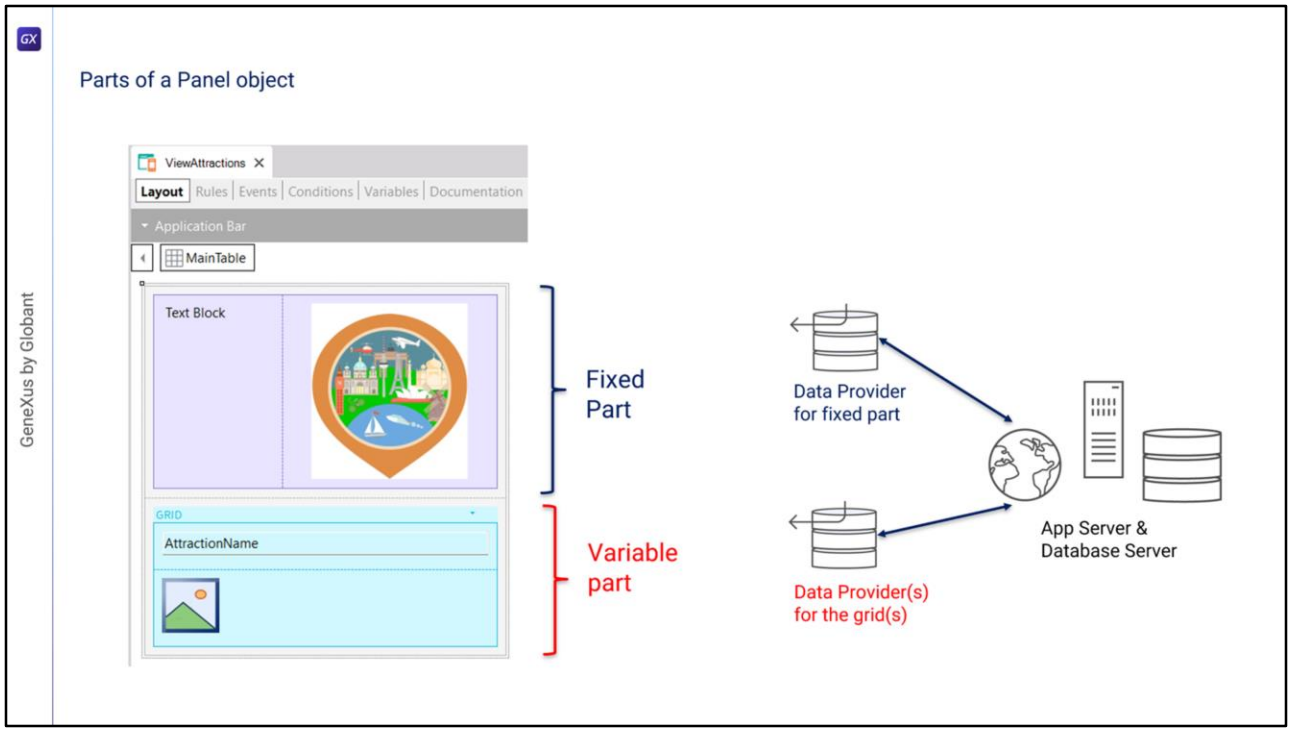
Los objetos Panel se diferencian de los WebPanels en que la parte fija se carga en forma independiente de la parte variable.



Antes de continuar vayamos a GeneXus y agreguemos algunos controles a la parte fija del Panel ViewAttractions que habíamos creado. Arriba y afuera del grid, agregamos una tabla con un Textblock de nombre TextblockTitle, alineación horizontal centrada y alineación vertical media, y una imagen (la misma que estamos usando en el menú, ya que no nos interesa aquí entrar en el diseño de la pantalla), con alineación horizontal centrada. Configuramos la propiedad Columns Style de la tabla para que la primera columna -que contiene al Textblock- ocupe un 80% del espacio total, y la segunda columna -que contiene a la imagen- el 20% restante.

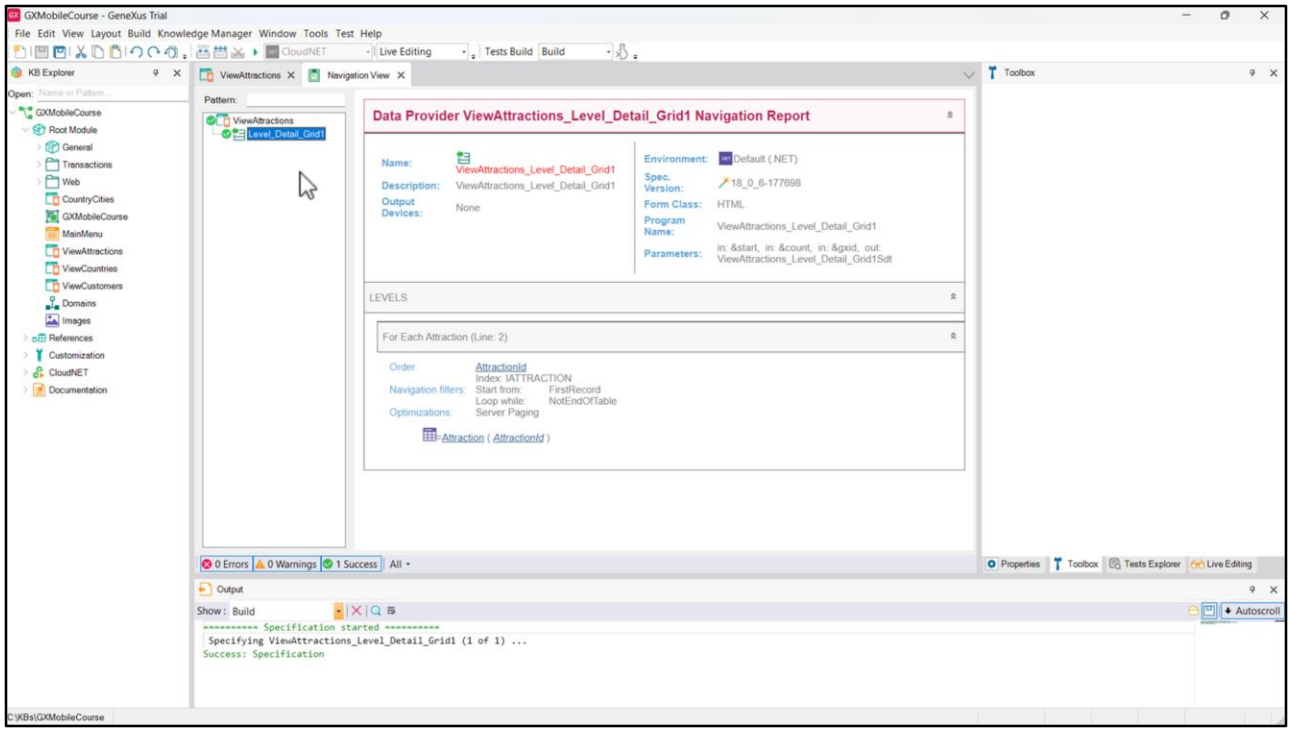


En el evento ClientStart, asignamos a su propiedad Caption con el título que queremos ver en pantalla, "The new age of EXPLORATION", salvamos y continuamos con el análisis de la carga de los datos del Panel.



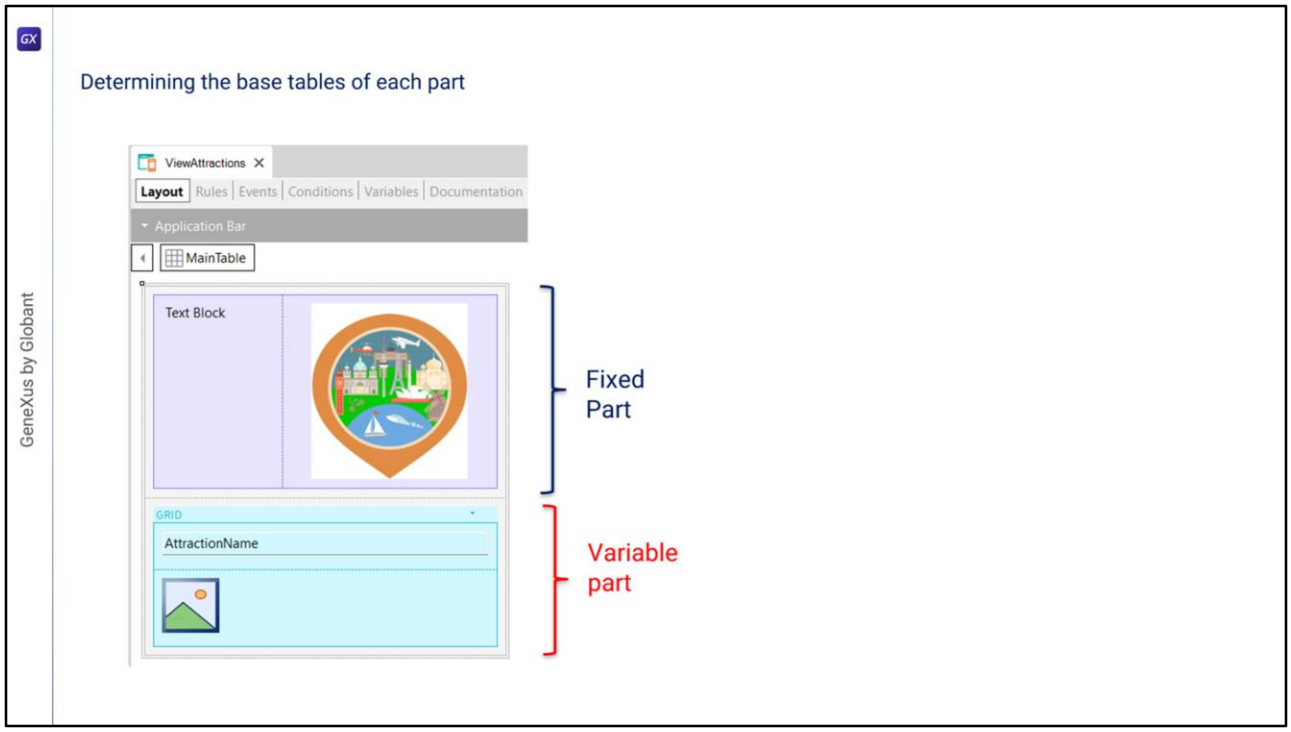
Para cada parte (fija y grid) GeneXus va a generar automáticamente Data Providers independientes que serán publicados como servicios en el servidor y serán los que accederán a la base de datos para obtener la información necesaria para cargar la parte fija y la parte variable del Panel.

Estos Data Providers no los veremos en la Base de Conocimiento ya que GeneXus se encargará de generarlos y mantenerlos, pero sí podremos ver a qué datos acceden si vemos su listado de navegación.



En el listado de navegación del Panel ViewAttractions vemos que bajo el nodo correspondiente al Panel aparece una entrada de nombre Level_Detail_Grid1, que corresponde a su parte variable, que en este caso está compuesta por el Grid1.
Si el Panel hubiera tenido más de un grid, aparecería un nodo de detalle por cada grid, ya que cada uno tendrá una navegación propia.

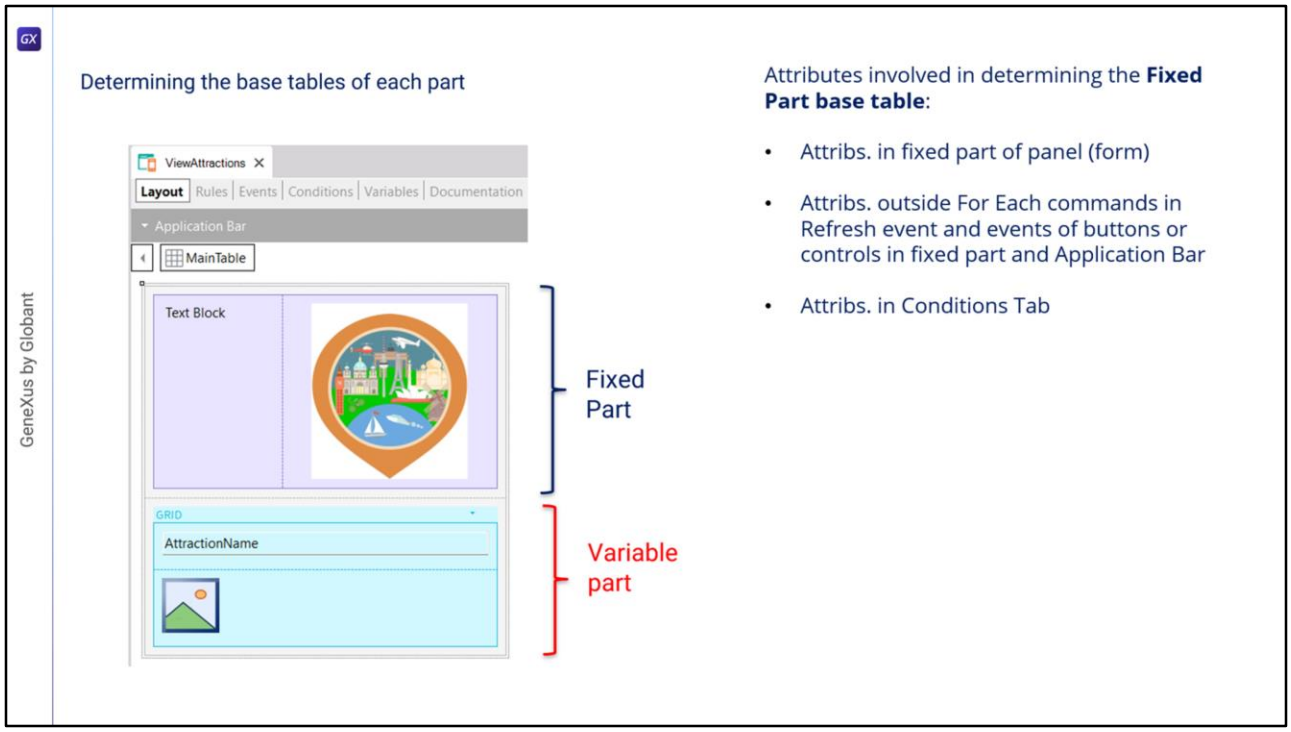
Seleccionamos entonces el nodo Level_Detail_Grid1, y vemos que recupera los datos del Data Provider ViewAttractions_Level_Detail_Grid1.
Este es el Data Provider que se creó automáticamente para cargar el Grid1, publicado como servicio en el servidor e invocado por el Panel para acceder a la base de datos y recuperar las atracciones.
Si reparamos en donde dice Environment, observamos que el Data Provider fue generado en .NET como parte del código del backend.
En la sección de parámetros, vemos que el Data Provider recibe algunos datos del grid y devuelve cargado un SDT de nombre View_Attractions_Level_Detail_Grid1, que contendrá los datos para cargar el grid.
La información del listado es similar a la que veríamos con un Web Panel con tabla base Attraction, ya que la grilla del Panel está recorriendo la tabla Attraction para mostrar las atracciones turísticas.



Dado que la parte fija y las partes variables del Panel se cargan en forma separada, las tablas bases de cada parte son completamente independientes entre sí. Conocer cómo GeneXus determina qué tablas debe acceder y la forma en que las recorre, es vital para que nuestra aplicación funcione correctamente.

En el ejemplo vemos elementos en la parte fija del Panel y un grid, por lo que se determinarán las tablas base de la parte fija y de la parte variable por separado. Puede darse el caso de que la parte fija tenga tabla base y el grid no, que el grid tenga tabla base y la parte fija no, que ambas partes tengan tabla base o que ninguna parte la tenga.

Es importante recordar que en esto los Panels funcionan distinto que los WebPanels, ya que en un WebPanel con un solo grid, en caso de que haya tabla base, la tabla base del WebPanel es única y no hay dos tablas bases distintas separadas para parte fija y grid, como sucede con los Panels.

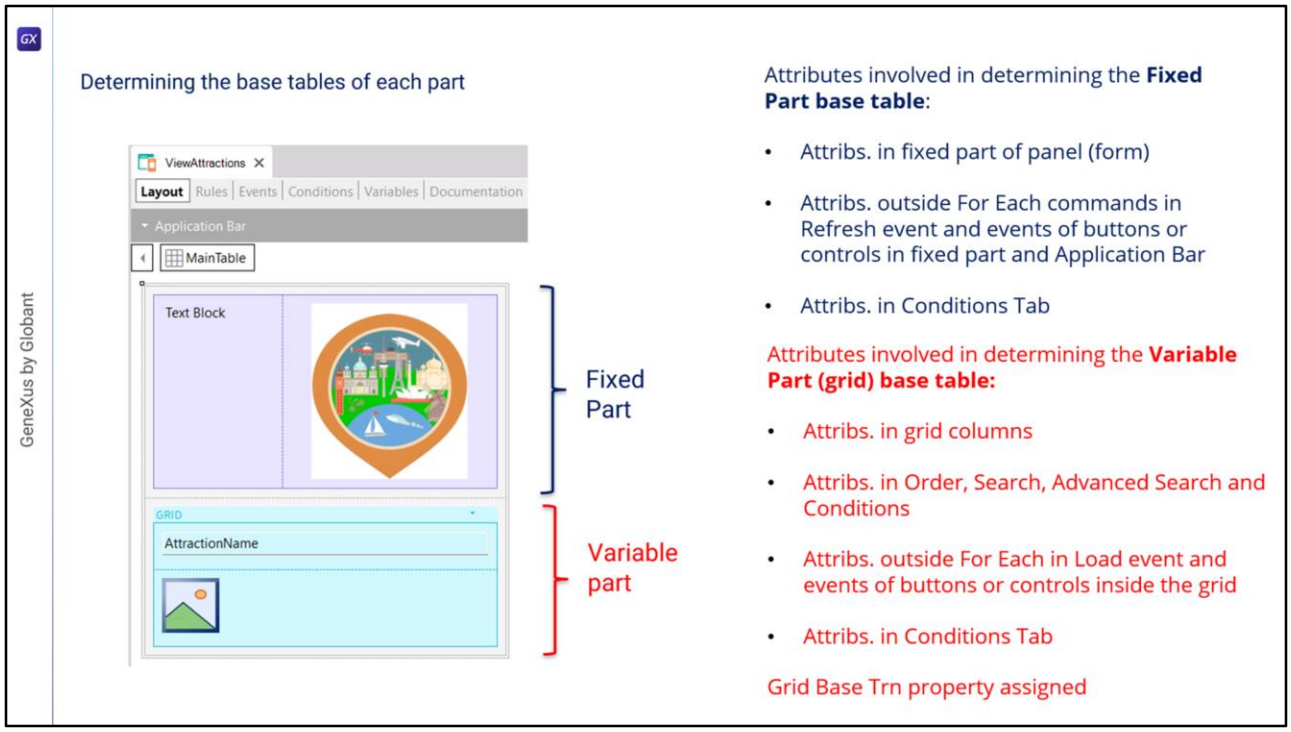


Dado entonces que en un objeto Panel la parte fija y el grid determinan navegaciones independientes y cada parte tendrá su tabla base, es como si hubiera dos for eachs paralelos.

Para determinar la tabla base de la parte fija, se tendrán en cuenta los atributos que pertenezcan a la parte fija del form y los atributos que pertenezcan a los eventos asociados a la parte fija, siempre y cuando estos atributos estén fuera de un comando For Each.

Estos eventos son el evento Refresh y los eventos asociados a botones o controles de la parte fija, incluidos los del Application Bar.

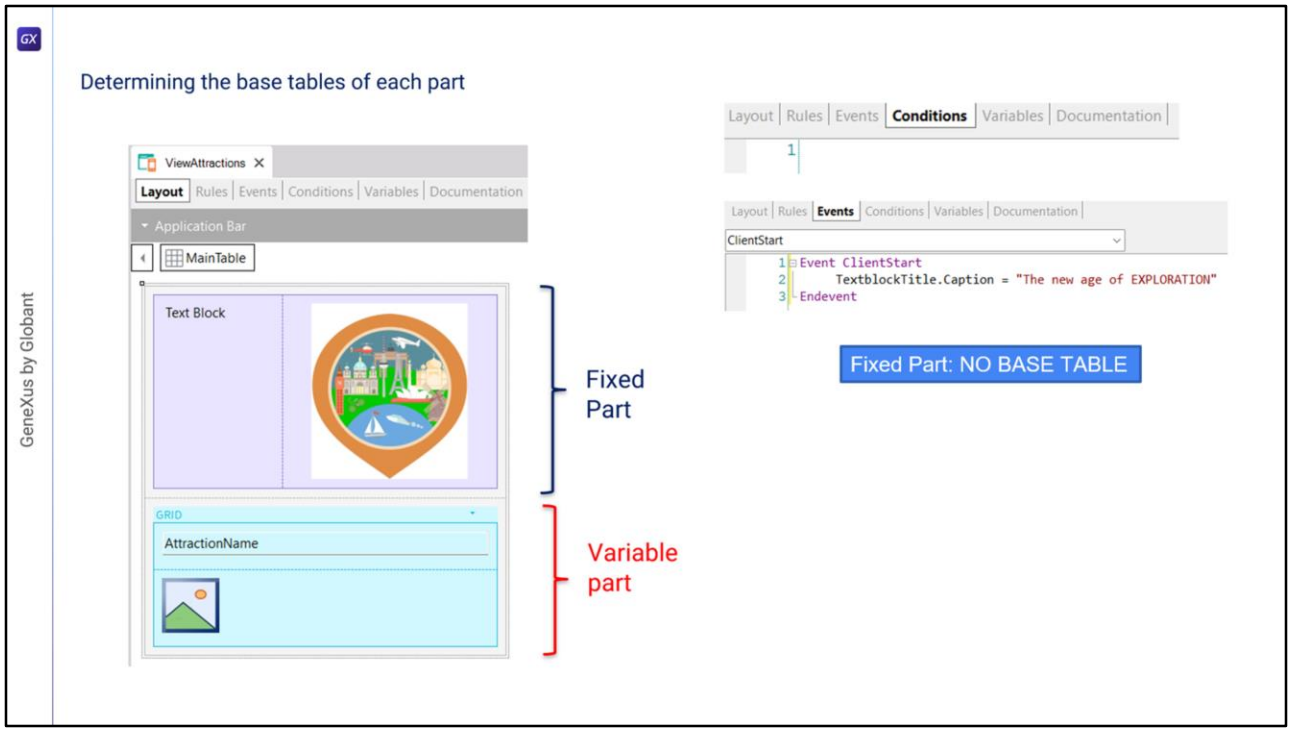
Además, para determinar la tabla base de la parte fija también se deben considerar los atributos del Tab Conditions del objeto Panel.



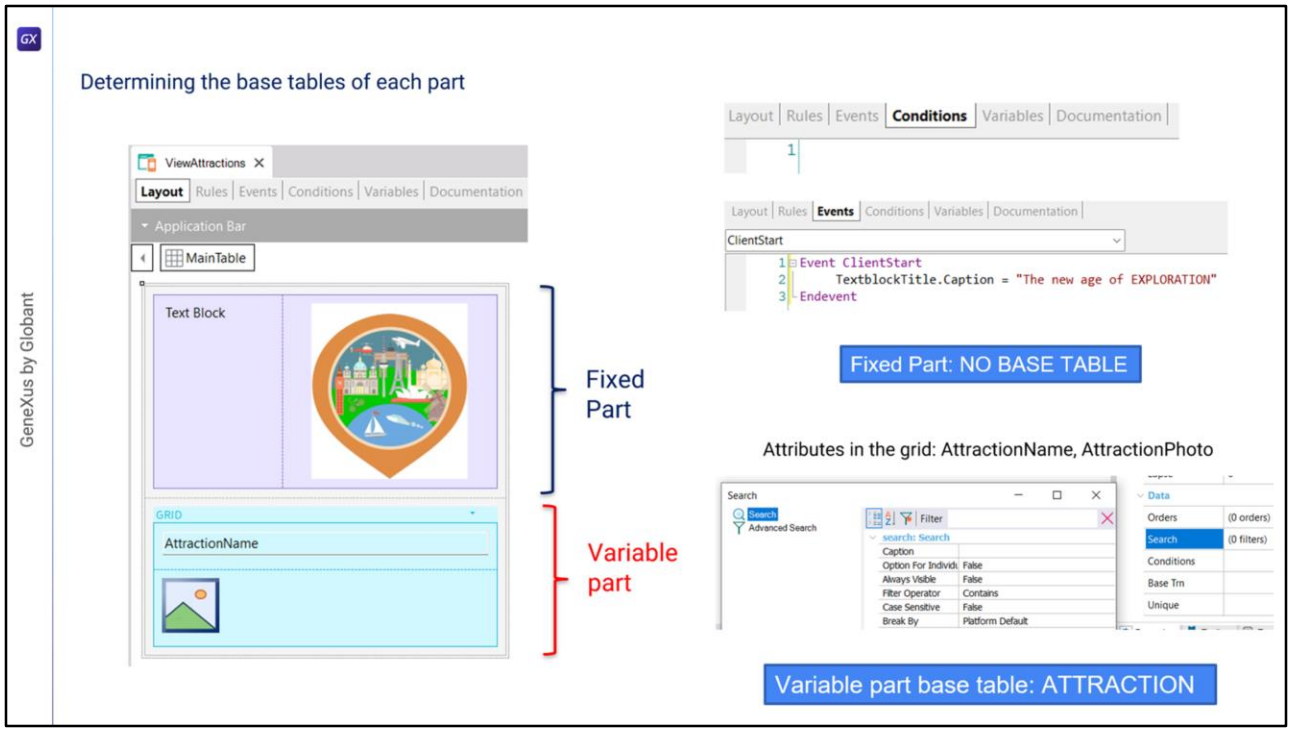
Para determinar la tabla base de la parte variable, en este caso del grid, se tendrán en cuenta los atributos incluidos en las columnas del grid, tanto visibles como no visibles, los atributos referenciados en el Order, Search, Advanced Search y Conditions del grid, los atributos que pertenezcan al código del evento Load siempre que estén fuera de cláusulas For Each y presentes en los eventos de botones o controles dentro del grid.

También se tomarán en cuenta los atributos que estén en el tab Conditions, que serán considerados para la determinación de la tabla base de todos los grids que se incluyan en el Panel.

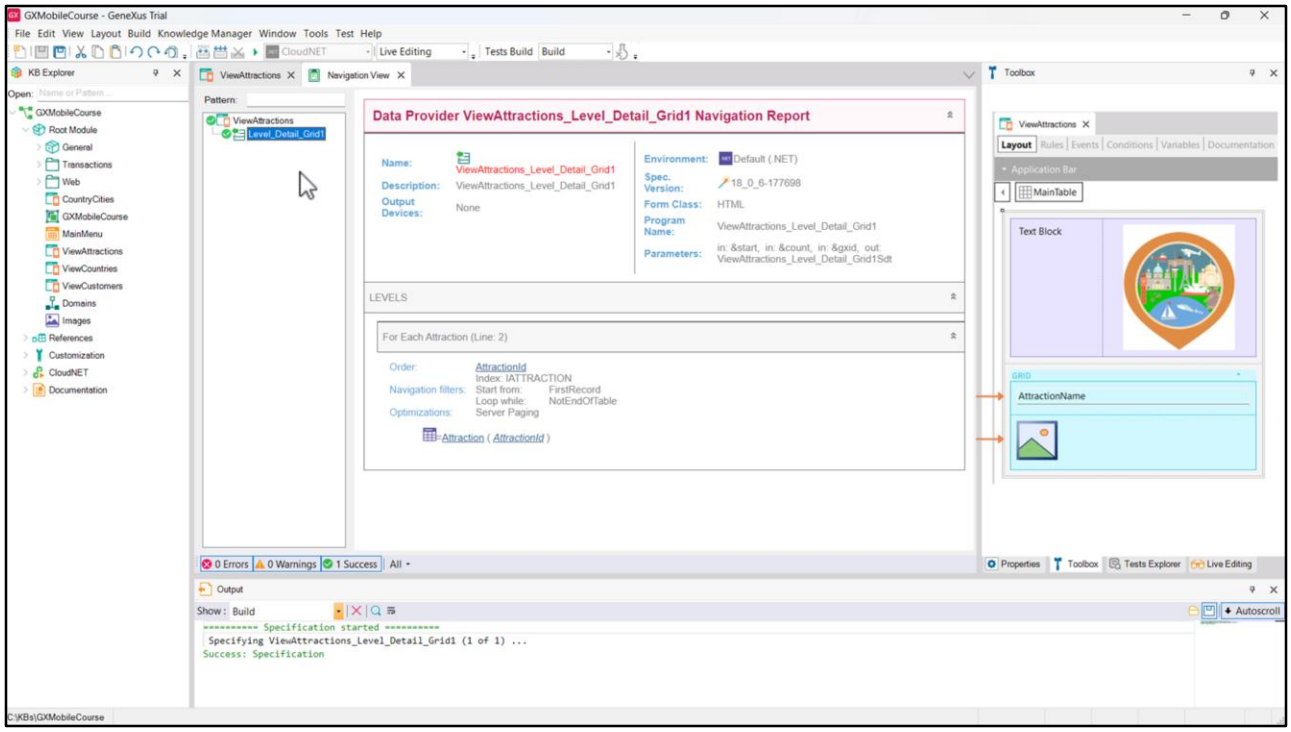
Por último, el grid también tendrá tabla base si su propiedad Base Trn fue asignada con una transacción base. En este caso los atributos que estén en las otras partes deberán pertenecer a la tabla extendida de la tabla asociada a la transacción base.



En el ejemplo que vemos, la parte fija está compuesta únicamente por una tabla, un texto y una imagen en el form, o sea que no hay atributos. Tampoco hay atributos en el tab Conditions del Panel. No hay evento Refresh ni tampoco eventos de controles del form o botones en la ApplicationBar, es decir que la parte fija de este Panel no tiene tabla base.

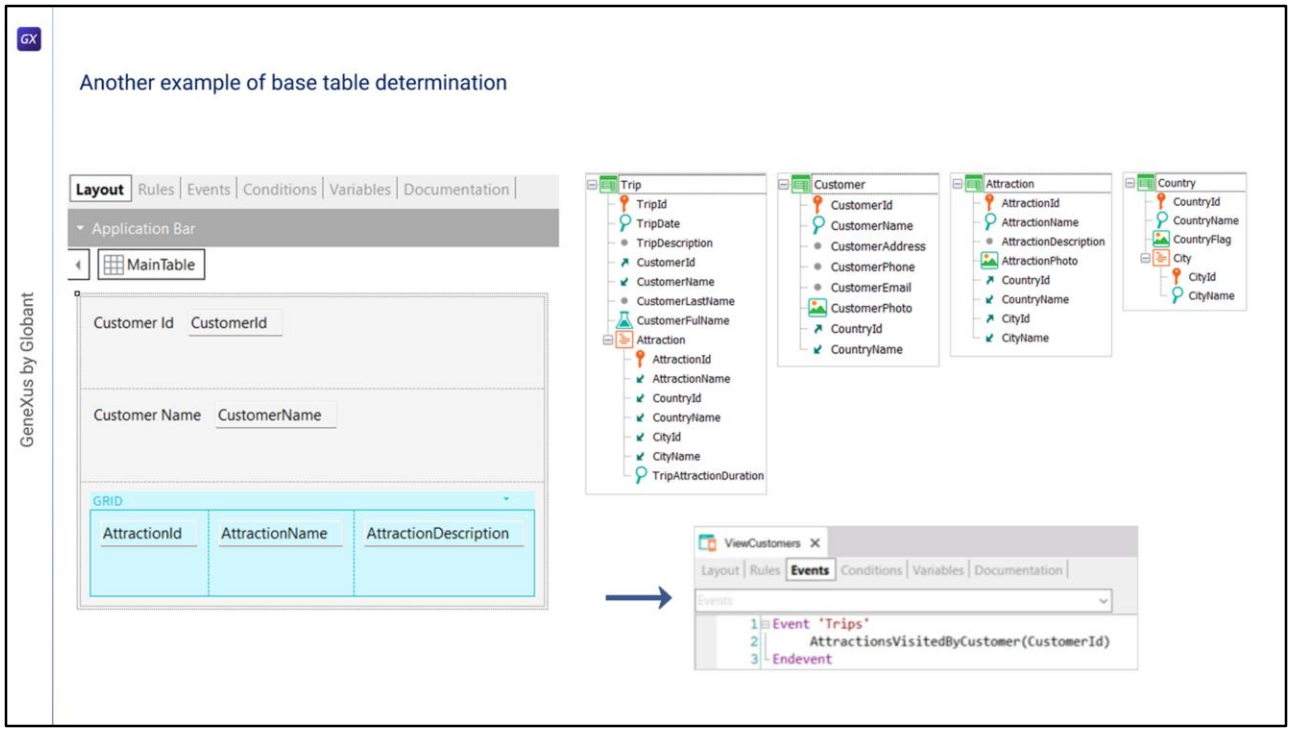


Si analizamos la parte variable, vemos que hay atributos en las columnas del grid, que son: AttractionName y AttractionPhoto. No hay atributos en las propiedades Order, Search, ni Conditions del grid y tampoco está asignado el valor de la propiedad Base Trn, y no hay atributos en el evento Load ni en eventos de controles dentro del grid. Por lo tanto, la parte variable del Panel, compuesta por el grid, tendrá tabla base Attraction. Esto significa que GeneXus construirá un Data Provider como servicio en el backend que recorrerá la tabla Attraction y recuperará los datos que cargarán la grilla.



Veamos el listado de navegación del Panel para constatar cuál es la tabla base de su parte variable: dice For Each Attraction, por lo que la tabla base es efectivamente ATTRACTION, como lo habíamos deducido y mencionado previamente.
Más abajo vemos que accede a la tabla Attraction para recuperar los datos de AttractionName y AttractionPhoto.

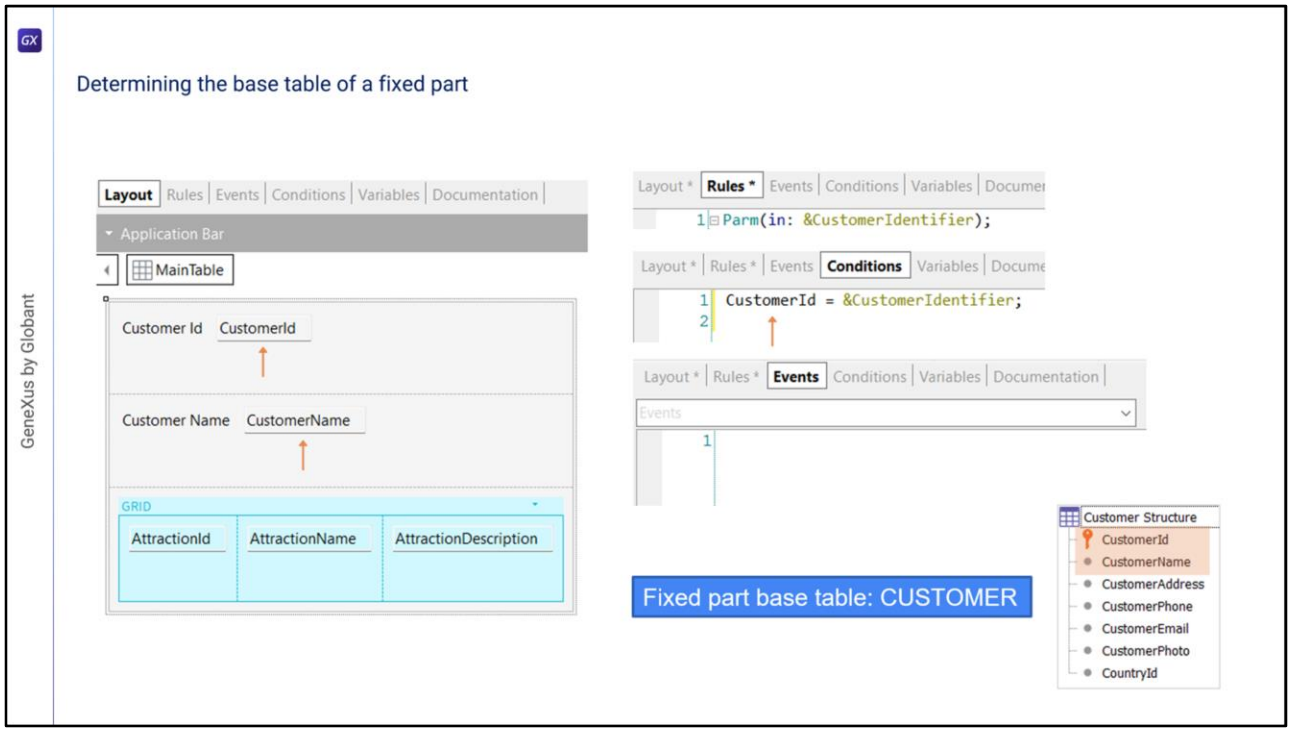
En resumen, vemos que por el solo hecho de que el grid tenga atributos, en este caso en sus columnas, GeneXus fue capaz de determinar automáticamente qué tabla debía recorrer y crear todo lo necesario para recuperar los datos requeridos desde la base de datos.



Veamos otro ejemplo de determinación de tablas base. La agencia de viajes quiere tener un Panel donde para un cierto cliente recibido por parámetro, se muestren las atracciones visitadas por ese cliente.

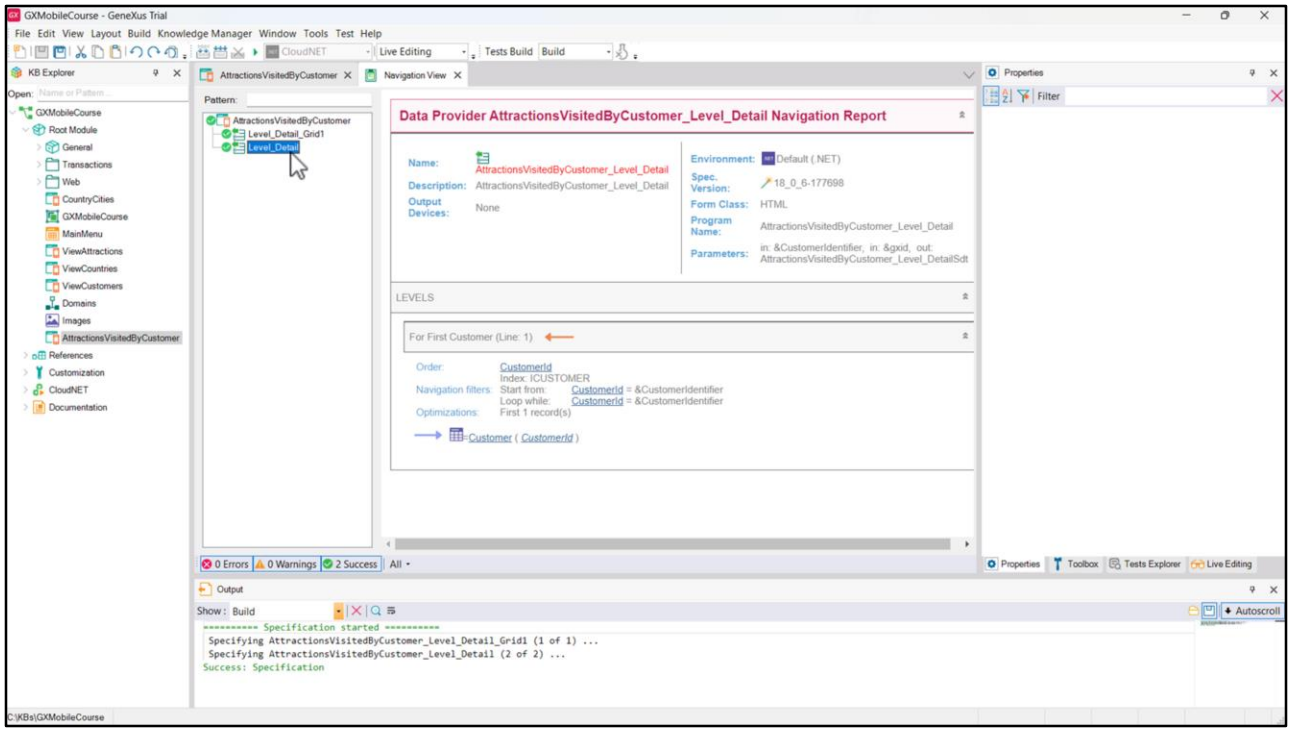
Esa información está modelada en la transacción Trip donde cada viaje tiene un cliente (observemos que CustomerId es clave foránea y el atributo CustomerName es inferido). Cada viaje tiene también muchas atracciones que son visitadas, representadas por el segundo nivel de la transacción Trip. También se muestran la transacción Customer que tiene los datos de los clientes, la transacción Attraction con los datos de las atracciones y la transacción Country con los datos de países y ciudades.

Para probar este ejemplo, invoca a este Panel desde otro objeto de tu KB, pasándole como parámetro el identificador del cliente.

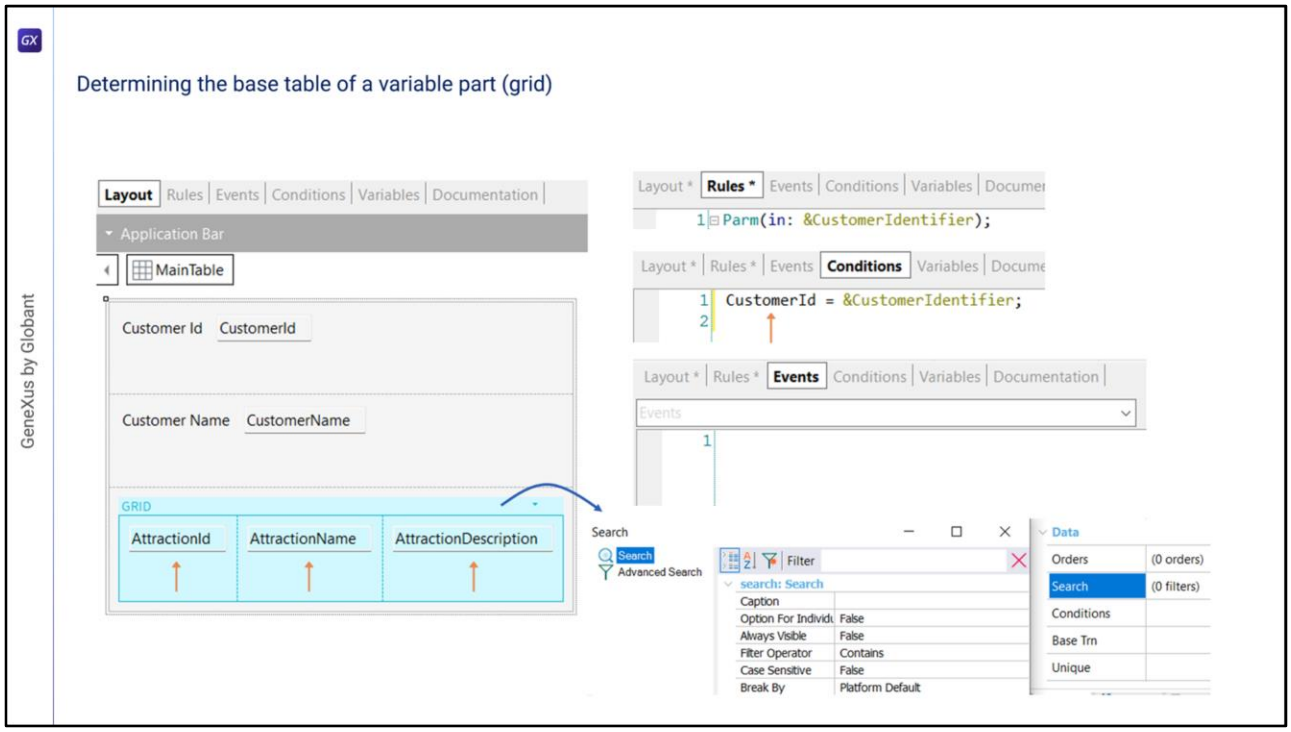


Analicemos primero la parte fija del Panel. Además de los atributos presentes en el form, si buscamos atributos en otras partes del Panel, vemos que en las reglas está solamente la variable &CustomerIdentifier que recibe el identificador del cliente a mostrar y en el tab Conditions encontramos un filtro que asegura que se mostrarán los datos solamente del cliente recibido por parámetro.
No tenemos nada en la solapa eventos.

Quiere decir que los únicos atributos que debemos analizar para la parte fija son CustomerId (presente en el form y en el tab Conditions) y CustomerName presente en el form.
La tabla base es entonces CUSTOMER pues contiene a ambos atributos.

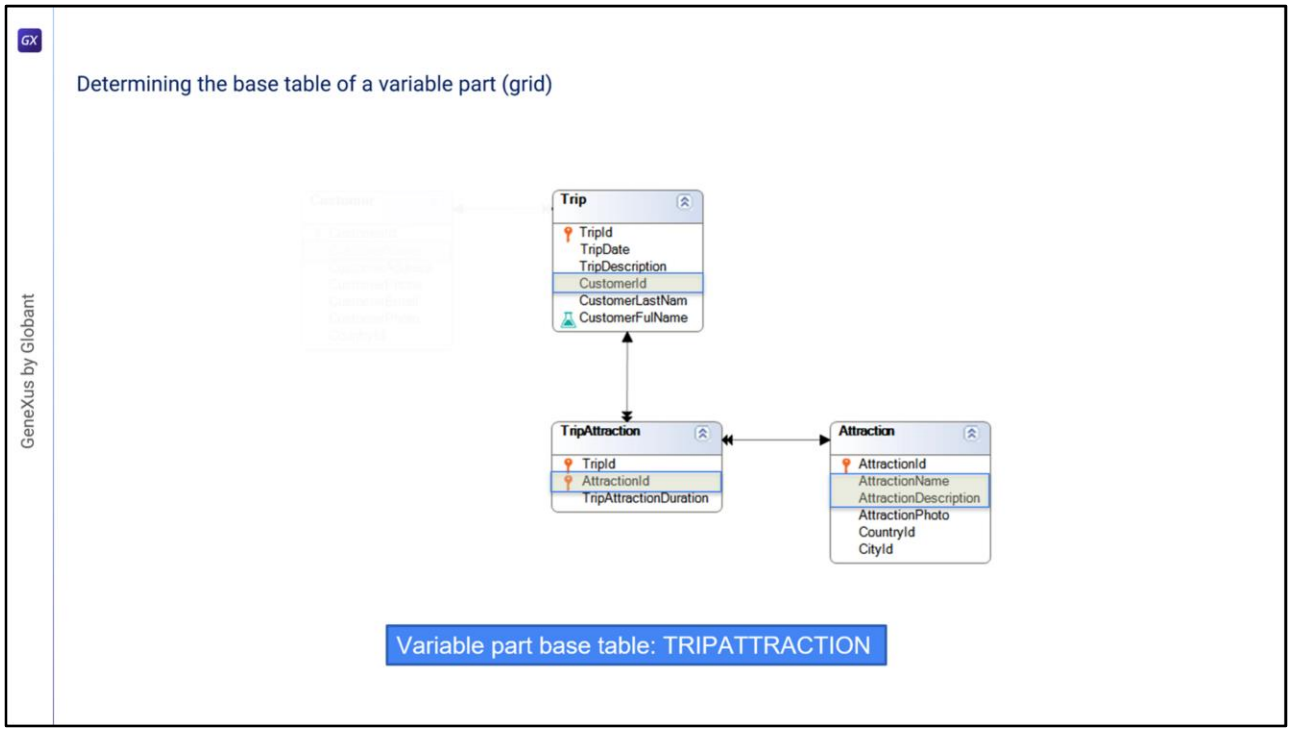


Si vemos el listado de navegación del nodo Level_Detail correspondiente a la parte fija, vemos que la tabla base es Customer, como habíamos determinado antes y que se accede a dicha tabla para recuperar los datos de los atributos CustomerId y CustomerName.

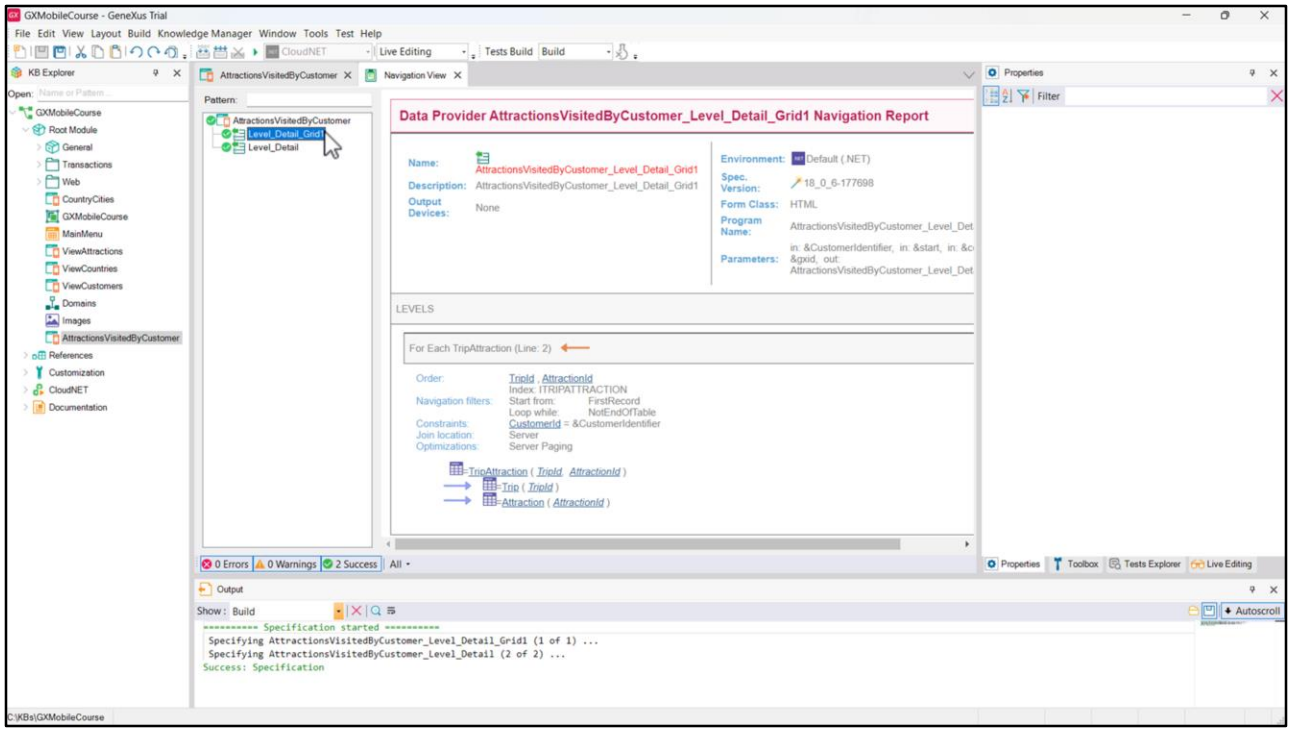


Si ahora analizamos la tabla base de la parte variable, tenemos a los atributos AttractionId, AttractionName y AttractionDescription como columnas del grid y no hay atributos en ninguna de las propiedades del grid.
Estamos tentados a afirmar que la tabla base del grid sería ATTRACTION, pero no nos olvidemos que los atributos del tab Conditions también deben considerarse y allí encontramos al atributo CustomerId como parte de un filtro.

Por lo tanto los atributos que encontramos son: AttractionId, AttractionName, AttractionDescription y CustomerId.



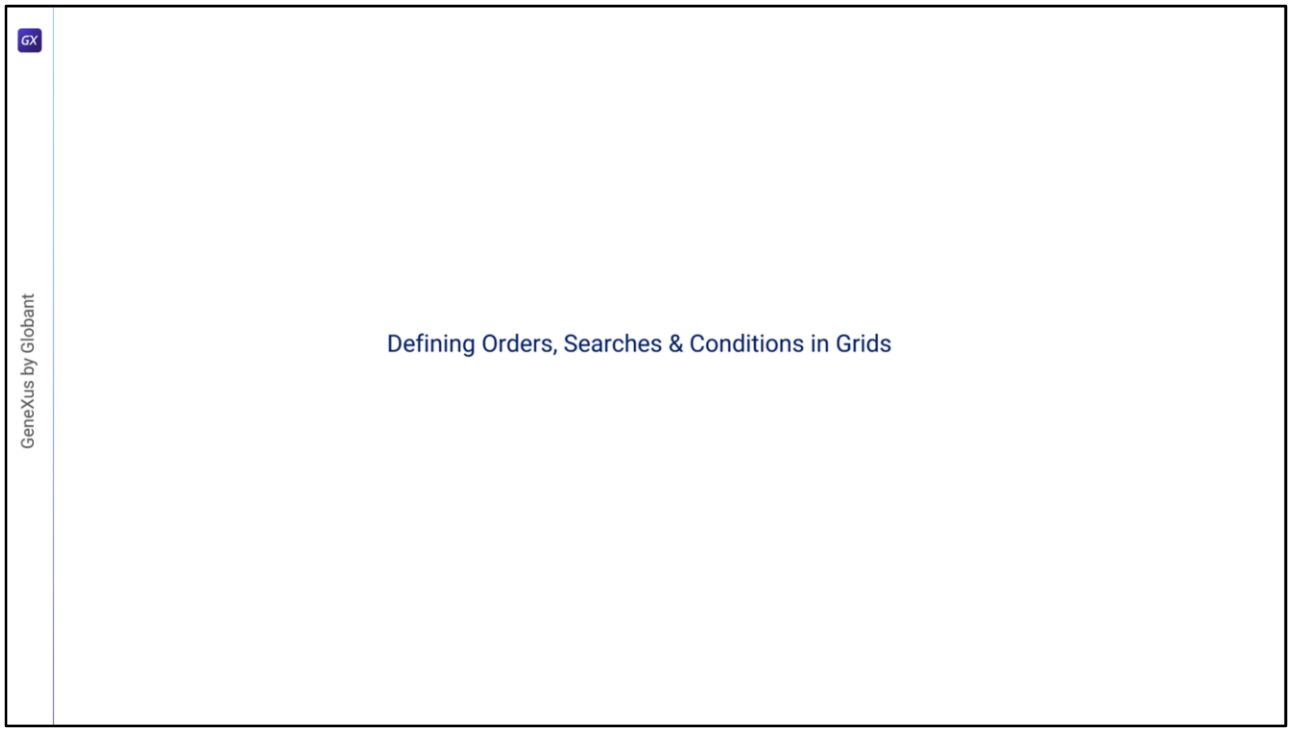
Si analizamos el diagrama de tablas, vemos que la única tabla extendida que contiene a estos atributos es la tabla extendida de TripAttraction, por lo que la tabla base de la parte variable del Panel será TRIPATTRACTION.



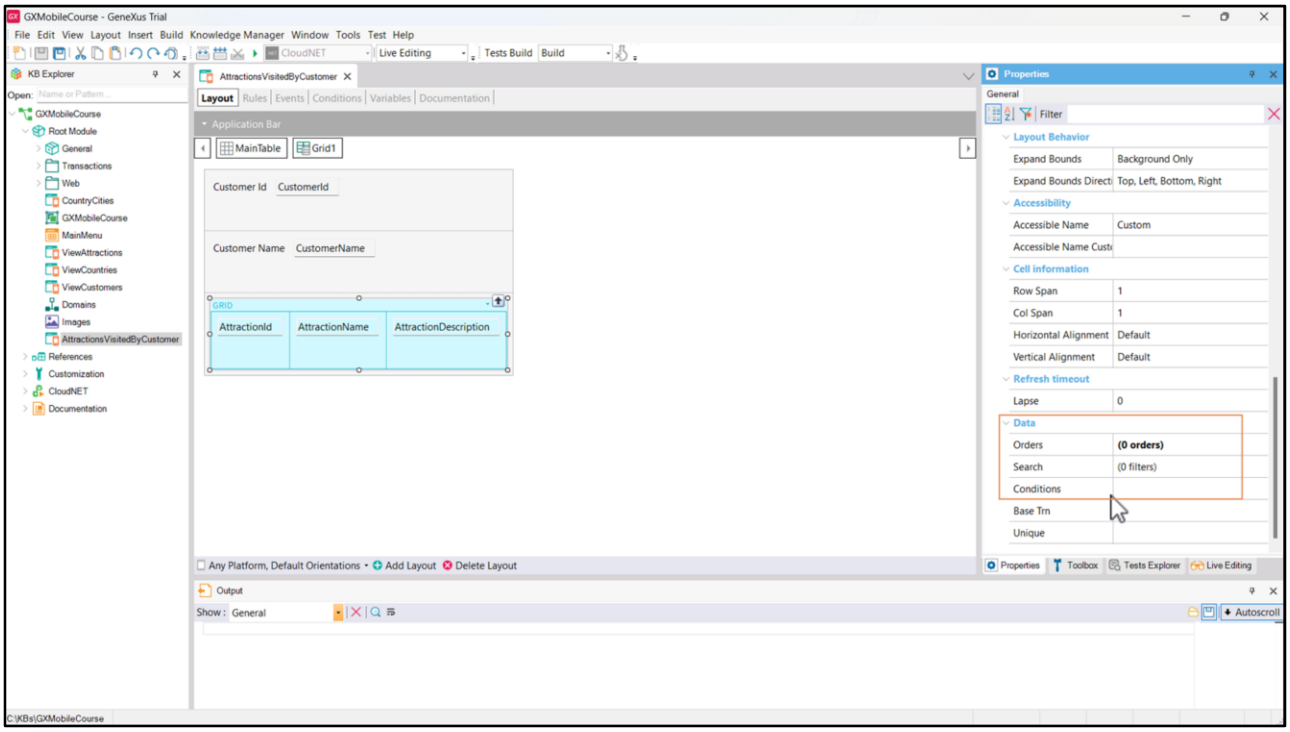
Si vemos el listado de navegación del nodo Level_Detail_Grid1 correspondiente a la parte variable, vemos que la tabla base es efectivamente TRIPATTRACTION, y que la misma está filtrada por el valor de CustomerId, debido a la condición presente en el tab Conditions.

También vemos que se accede a las tablas Trip y Attraction parar recuperar a los datos de los atributos CustomerId, AttractionName y AttractionDescription.

Anteriormente hemos mencionado que el código presente en los eventos también es tomado en cuenta para la determinación de las tablas base de la parte fija y parte variable. Veremos en otro video los eventos de un objeto Panel.

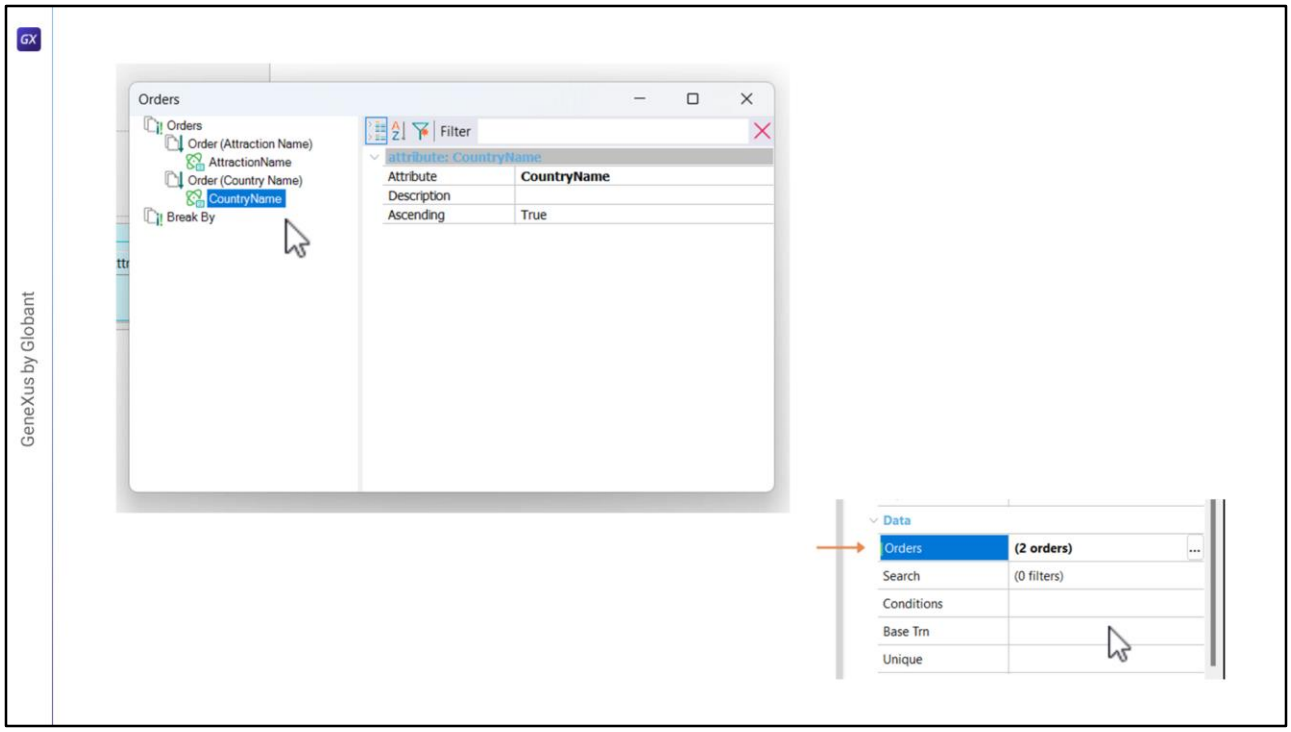


Veamos ahora cómo podemos ordenar y agrupar la información y cómo hacer búsquedas y establecer filtros y condiciones.



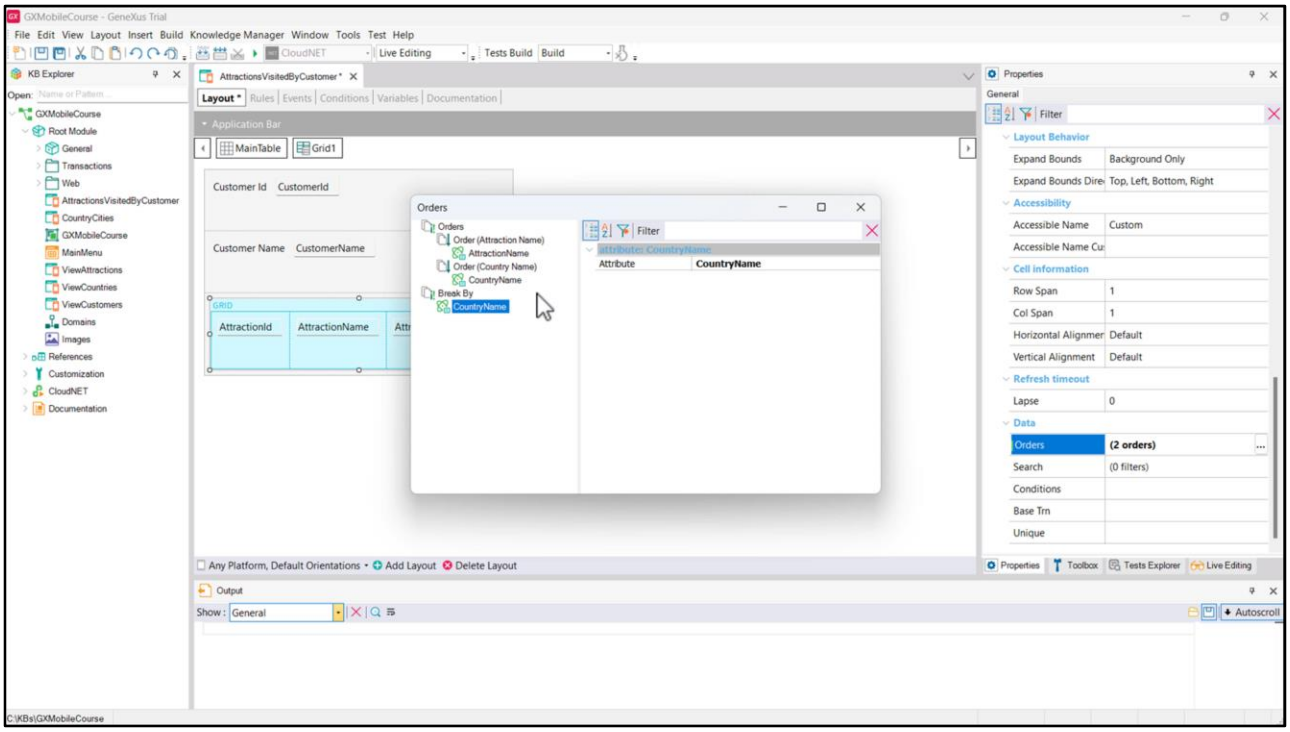
Una característica fundamental cuando presentamos información en grillas es que la misma aparezca ordenada, según uno o varios criterios, de manera de ayudar al usuario a procesar esa información. Podríamos, por ejemplo, querer que la información en la lista de atracciones aparezca ordenada por nombre o podríamos querer ordenarla por el nombre del país en el que se encuentran.

Si vamos a las propiedades bajo el grupo Data del grid, vemos que no fueron definidos órdenes, búsquedas ni condiciones para los datos.

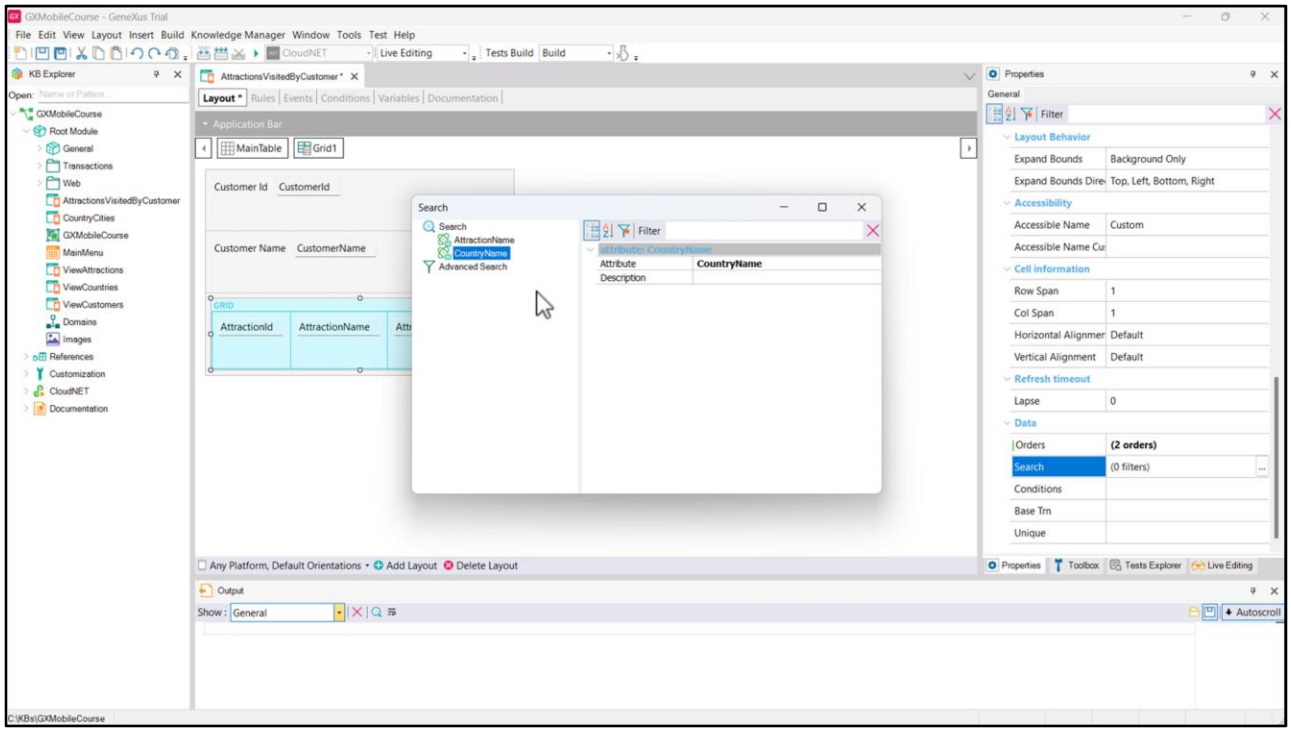


Para definir nuestros órdenes, abrimos el editor de la propiedad Orders, hacemos clic derecho, Add, Order, le ponemos el nombre, en este primer caso Attraction Name, y sobre él hacemos clic derecho, Add, Attribute: aquí elegimos al atributo AttractionName. Creemos, de la misma manera, el orden para ordenar por nombre de país: clic derecho, Add, Order, le ponemos el nombre Country Name… y agregamos al atributo CountryName.

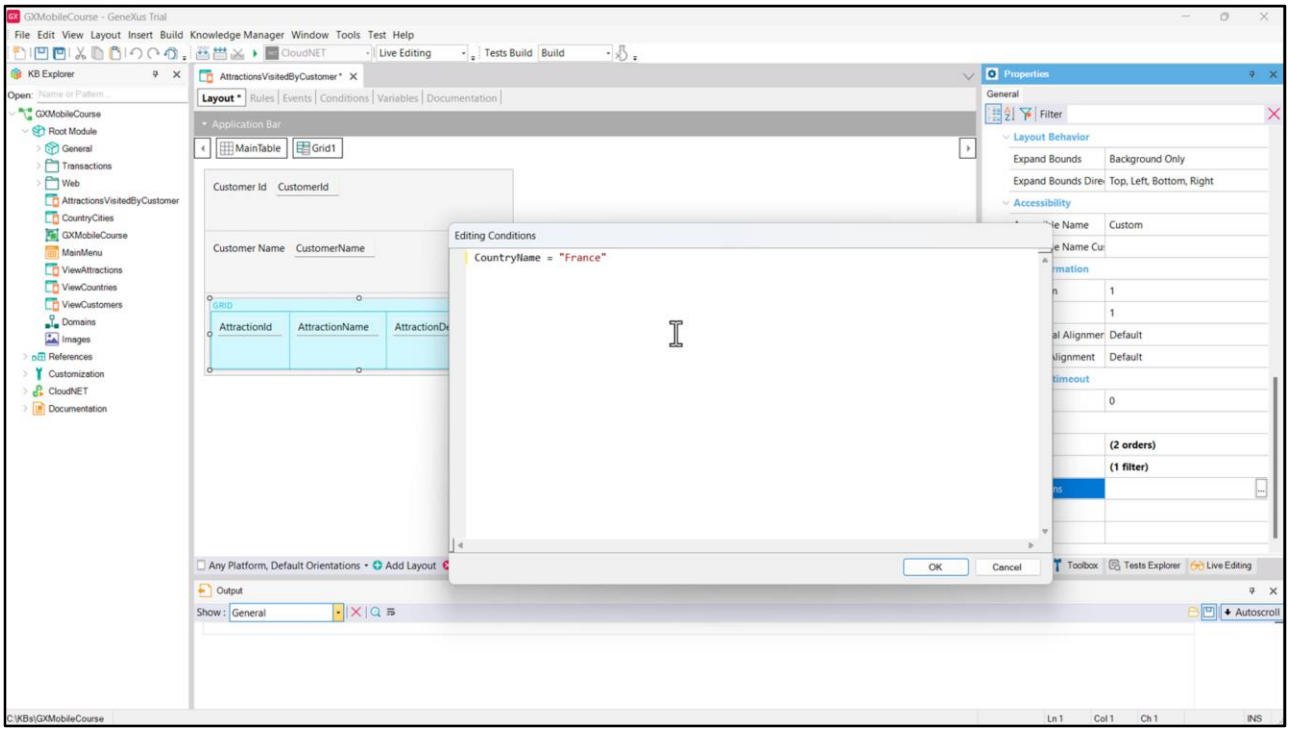
Vemos que ahora aparecen 2 órdenes definidos.



En lugar de querer ordenar por nombre de país, es posible que al usuario le interese también tener la posibilidad de agrupar las atracciones según el país al que pertenecen. Volvemos a la ventana de definición de órdenes, y en el nodo Break By hacemos clic derecho, Add, Attribute, y elegimos al atributo CountryName para que haga el corte de control.



Otra de las funcionalidades fundamentales que toda aplicación debe proveer son las búsquedas. Por ejemplo, podríamos pensar en la posibilidad de buscar atracciones por nombre o país. Para eso, abrimos el editor de la propiedad Search, y simplemente agregamos los dos atributos por los que queremos buscar: AttractionName y CountryName.



Además, ocasionalmente puede ser necesario que algunos registros cumplan ciertas condiciones para ser mostrados: podríamos querer que la grilla de atracciones muestre únicamente atracciones que estén en Francia, por ejemplo, entonces en la propiedad Conditions, escribimos: CountryName = ‘‘France’’.

En este video estudiamos la lógica que emplea GeneXus a la hora de cargar los datos en la pantalla de un Panel y determinar sus tablas base, y vimos cómo mejorar la experiencia de los usuarios de nuestra aplicación ofreciendo órdenes, búsquedas y condiciones para los datos a ser mostrados.